V o c a l c o m

W e b S c r i p t e r U s e r M a n u a l

Version	Date	Author
v3.0	09/06/2005	Vocalcom Benelux

Index

1 INTRODUCTION 6

1.1 General Information 7
1.2 Who should read this documentation? 7
1.3 Application domain 8
1.4 Operating Principles 8

2 SETUP 10

2.1 Prerequisites for the client station 11
2.2 How to improve client station performances 12
2.3 Launch 13
2.3.1 The applications selection window 16

3 SCRIPT MANAGER 18

3.1 Scripts List 19
3.2 Create a new script 22
3.3 Restore scripts 22

4 THE SCRIPT EDITOR 24

4.1 Editor menus 25
4.1.1 The main menu 26
4.1.2 The Objects Menu 37
4.2 The Edition Zone 43
4.2.1 Select an Object 44
4.2.2 Object Groups 45
4.2.3 Help dialog windows 47
4.3 Navigation Tabs 53

5 OBJECTS AND PROPERTIES 55

5.1 Property Overview 56
5.2 The Objects 67
5.2.1 PAGE Object 67
5.2.2 LABEL Object 71
5.2.3 TEXT Object (TEXTBOX) 78
5.2.4 TEXT Zone Object (TEXTAREA) 82
5.2.5 BUTTON Object (BUTTON) 84
5.2.6 LINK Object (HYPERLINK) 87
5.2.7 IMAGE Object 89

5.2.8 DROP-DOWN LIST Object (DROPDOWNLIST) 90
5.2.9 LIST Object (LISTBOX) 95
5.2.10 CHECKBOX Object 96
5.2.11 RADIO Object (RADIOBUTTON) 98
5.2.12 OBJECTION Object 99
5.2.13 The IFRAME Object 103
5.2.14 DATAGRID Object 104
5.2.15 The LITERAL Object 108
5.2.16 The CALENDAR Object 109

6 COMPOSITION OF A SCRIPT 112

6.1 How to generate a script 113
6.1.1 Create a new script 113
6.1.2 How to manage script pages 114
6.1.3 Page layout 115
6.1.4 Create objects on the script page 115
6.1.5 Text format via cascading style sheet 117
6.1.6 Text zones and lists 117
6.1.7 Links and buttons 120
6.1.8 Images and IFrames 120
6.1.9 Checkboxes and radio buttons 122
6.1.10 Datagrid 122
6.1.11 The calendar 123
6.1.12 Create connections with databases 124
6.1.13 Create the Client file 126
6.1.14 Save and execute a script 128
6.1.15 Debug Window 129
6.2 Upload Center 135
6.3 Link Builder 136
6.4 Criteria Builder 140
6.5 Connection Link 142
6.5.1 Connection Auto Link : Start Customer File Connection 144
6.5.2 Connection Create Link : Start Customer File Creation 146
6.5.3 Invalid connections check at script opening 148
6.6 Connection Viewer 149
6.7 Connection Builder and Checking 150
6.8 Actions and Commands 152
6.8.1 Description of the possible commands 157
6.8.2 Objects References 160

6.8.3 Definition of global variables 160
6.9 Examples of useful scripts 164
6.9.1 Greeting message according the time 164
6.9.2 Use agent identification information 165
6.9.3 Change the mouse pointer 166

LEGAL STATEMENT

This documentation is protected by national and international copyright laws.

The VOCALCOM® name and its logo are registered trademarks of VOCALCOM S.A. with its corporate head office at 7 rue de Tilsitt, 75017 Paris, France. The HERMES PRO™ name is protected by national and international usage rights for a trade name and more generally, by national and international software protection legislation. The other names, brands and trade names mentioned belong to their respective owners.

Reproducing all or part of this documentation on any media whatsoever is forbidden without the express prior approval of the publisher (art. L122-4 and L122-5 of the French Intellectual Property Code).

The publisher cannot accept liability for any typographical errors, picture printing errors or any other symbol error nor can it accept liability for consequences arising from the incorrect use of this documentation.

The sole purpose of this documentation is to educate and train individuals. It cannot in any way be interpreted as a contract, agreement (including *sui generis*), an advertising and/or promotional media, in any form whatsoever.

© 2000-2005 VOCALCOM S.A. – All Rights Reserved

1 INTRODUCTION

Vocalcom Web Scripter™ is a client-server solution that enables call center administrators to develop and generate scripts.

As it was specifically developed for full-Web usage, it can be deployed in independent call centers, as well as in ASP mode. All the Web Scripter's functionalities can be accessed through an Internet browser, thus not requiring installation of an Active X component. Its web-enabled character renders the solution extremely user-friendly, flexible and scalable.

1.1 General Information

The Web Scripter™ has a user-friendly interface that contains all necessary elements to develop entire script pages that can easily be configured in a minimum of time. Its two different user modes, basic and advanced, allow a gradual evolution to mastering the product's most advanced features.

The Web Scripter™ allows you to link your script to your databases, either by creating a database with the editor, or by linking fields to your existing databases. The different pages of your script can be enriched with contextual connections, automatic actions or actions that are launched upon specific events.

The Web Scripter FrameSet Editor™ is independent from the main editor and enables you to create pages of the FrameSet type. You can organize your frames fast and with ease of use thanks to its very graphical nature, thus creating complex and structured script pages without any difficulty.

In short, the abundant and graphical features of the Web Scripter provide unrivaled ergonomics and functionalities.

1.2 Who should read this documentation?

This documentation is aimed at all Web Scripter™ users, so basically contact center administrators who have to develop scripts. We enlarge upon all aspects that are useful for generating scripts.

Among the key elements of this documentation are: creating visual, attractive-looking pages, creating simple and complex links between pages, creating commands and actions, and linking scripts with your databases.

1.3 Application domain

Although the Web Scripter™ can be used to create any kind of screen script, some of its integrated functions and commands make it perfect for usage in combination with Vocalcom Hermès Tools™ or Vocalcom Hermès Pro™.

Thanks to its fully web-enabled character, the Web Scripter™ can easily be integrated with any application type as it merely requires an Internet browser to be operational. It can be accessed through a specific URL.

One of the major advantages of the web-based solution is its use in delocalized or multiple sites through a thin client such as a LAN (Local Area Network) or a WAN (World Area Network).

1.4 Operating Principles

The Web Scripter™ is based on so-called Client-Server technology:

Both the application and all information with regard to data registered by its users, reside on a single machine - the Server - thus centralizing everything in one location.

Users can deploy the application from any workstation (the Client) on the condition that they can connect the Server that houses the application. Depending on the circumstances, the user can connect through the company network or the Internet.

Because of its shared architecture, all servers equipped with the Web Scripter™ application can communicate with each other through Web Services. This functionality allows importing and exporting of scripts between machines. This is valid for deployment both in Editing as well as in Production mode.

In the Editing mode, the administrator creates his script through an Internet browser while all information and modifications are stored, in real time, on the Server.

In Production mode, the agent browses the script with his application (Web Agent™, Hermès Tools™ or another application that can be integrated with Web Scripter™ by launching an Internet browser).

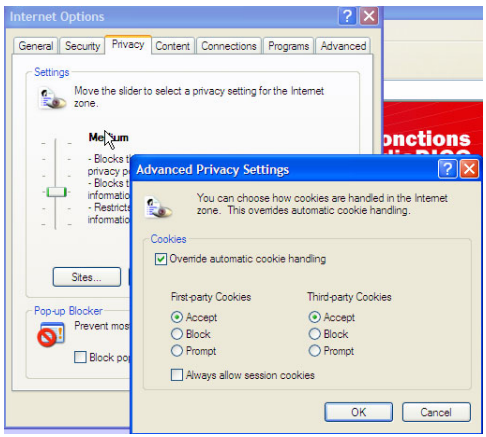
2 SETUP

This chapter provides general information with regard to the installation and start-up of the application.

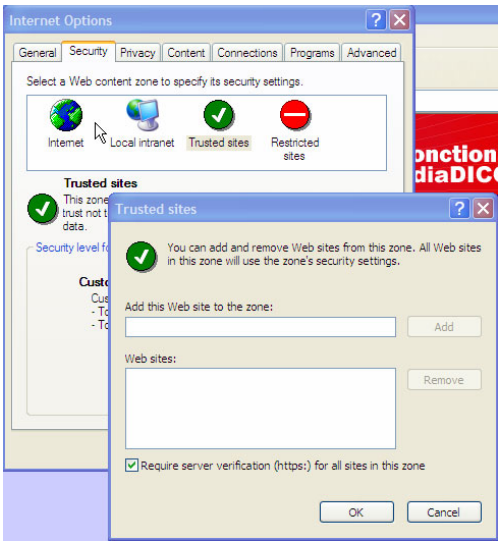
2.1 Prerequisites for the client station

The WebScripter V3 application must be launched with Internet Explorer, minimum version 5.5. Certain configurations might prevent the Web Scripter from starting up correctly on the client station. 'PopUp killer' applications (Google bar, Norton, etc.) prevent pop-ups to open within a browser, thus they should be deactivated.

Certain anti-virus (Norton, etc.) programs can be configured in a way that they prevent the execution of Java scripts in browsers. As your browser must accept cookies, go to your Internet Explorer's Options, click on the 'Privacy' tab and select the following options:



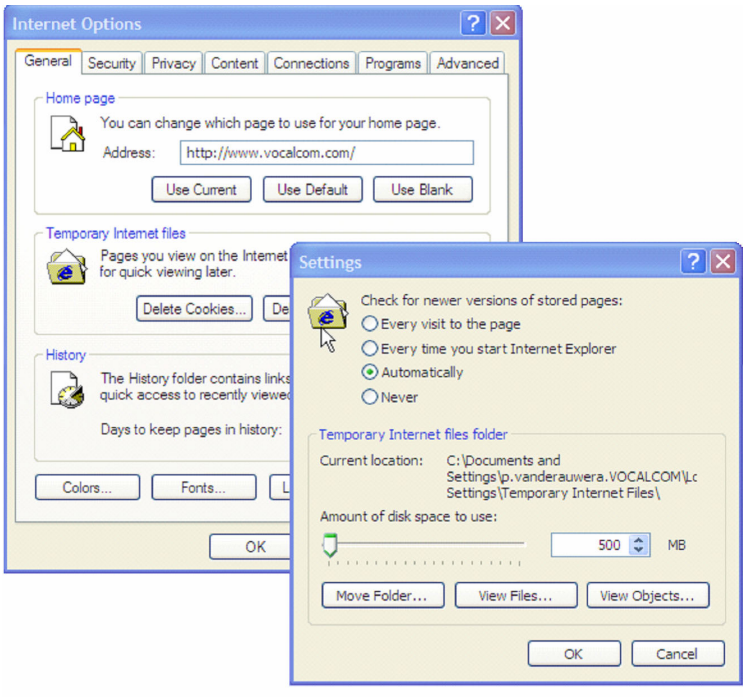
You may have to add your IIS server's address to the trusted sites of your browser. In order to do so, go to the Internet Options, click on the 'Security' tab and select the following options:



Enter your web server's address, for instance:
http://My_Server. Then put the cursor 'Security Level' on low.

2.2 How to improve client station performances

The WebScripter V3.0 application requires 512Mo ram minimum to function properly. You can improve the application's performances by changing the size of your Internet files folder. Go to the Internet Explorer Options menu, click on the 'General' tab > 'Temporary Internet Files' > 'Settings' and select the following options:



You can obtain improved performances if you deactivate the ‘Check Java script’ option of your anti-virus system.

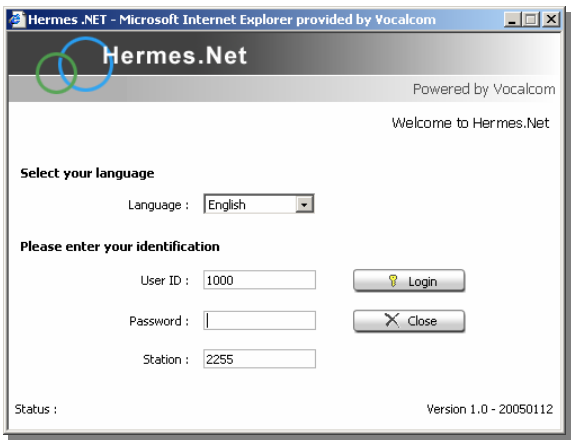
2.3 Launch

The Web Scripter™ is launched through an Internet browser, so launching your Internet Explorer and typing in the correct URL is sufficient to start the application.

Example of a URL to be launched in LAN (Local Area Network) mode:
http://name_iis_server/web_scripter

Example of a URL to be launched in WAN (World Area Network) mode:
http://ip_public_iis_server/web_scripter

« Add to your favorites or Internet shortcuts on the desktop »



The window used to identify the user appears.
The drop-down list '**Language'** allows you to select the editing language of your choice.

The

button can be used to close in one step all the opened Hermès.Net applications, together with this window.

Once the information is encoded, the user can press **Enter** or click on the **Login** button. The system checks login information. If the login exists and the password is correct, the next window appears: the **Applications selection** window.

Remark : if you add this URL in the list of trusted sites in the I.E. options of the computer, the status bar at the bottom of the screen won't appear anymore.

2.3.1 The applications selection window

This window allows launching one of the next applications:

Administration

- Web Scripter
- Agenda
- Reporting
- Supervision

These applications are available or not according to the rights access assigned to the identified user. If only one application is available, it's automatically launched. If more are available, the user must use the

button in front of the application he wants open.

At the bottom of the screen, the named of the user is showed. The

button can be used to come back on the previous window and change login information.

Only the Web Scripter application is described in this document. Please use other specific documents to get information about other applications.

« Please note that your ID is stored in a cookie on the client station (without the password). »

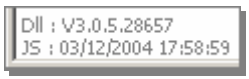
The dropdown list close to the application name can be used to select a specific site script.

3 SCRIPT MANAGER

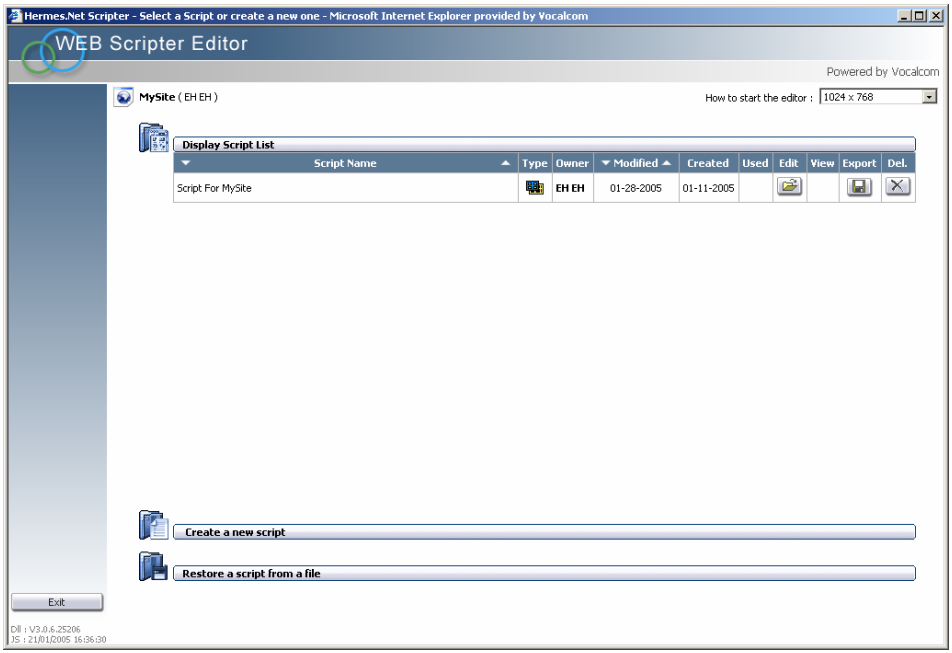
This chapter describes how to create a new script or how to identify and select an existing script on the script management screen. It also contains information on how to restore a script from a file.

Scripts available here belong to the user site/company.

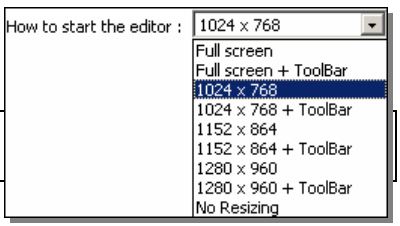
« dll and other file names are on the bottom of the screen; please write down this information when you call the Hotline. »



3.1 Scripts List



Remember: When installing your system, the administrator created a company (to which you belong) and an account for each user of your company. All throughout this documentation you will see that it is possible to share information between different users of the same company.



How to start the editor

The Web Scripter™ allows you to select the editing interface in the resolution of our choice. Simply select the screen resolution you wish to use from the list. You can also choose to display or hide the browser toolbar. The selected value will be associated with each startup.

Display Script List:

In this list you will see all the scripts created by yourself or someone else of your company. By default, the list is sorted by modification date (the most recent at the top). It's possible to sort the scripts on name or modification, descending or ascending order.

Script name: Identifier of the script

Type: 'Page Scripts': scripts created with the standard editor. 'Frame': scripts created with the 'FrameSet Editor'

Owner: Name of the person that did on the script last modifications.

Modified: Date of the latest modification.

Created: Date when the script was created.

Used: Indicates whether another user is already working on the script by means of a flickering « ! ». In function of the Web Scripter's configuration, you can prohibit opening a script that is already in use by someone else. For more information, check out the chapter

'Administrator Mode – Configuration files' of the administrator manual.

Once displayed, the script in use by multiple simultaneous users is indicated by the following on-screen reminder:

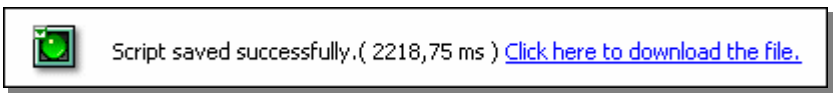
When a script is opened by 2 users, the last modifications done overwrite the previous modifications done by the other user. So to prevent loosing modifications, it's hardly disadvised opening an already opened script. It can be opened to consult only, but don't forget to uncheck the "Auto Save" option. (see section « Preferences Menu » on page 34)

Edit: Click on this icon to open the selected script.

« To open a script from the list, you can also double-click on its name. »

View: Click on this icon to execute the script in another window.

Export: Click on this button to export the selected script. Once exported, the next message is shown at the top of the scripts list:



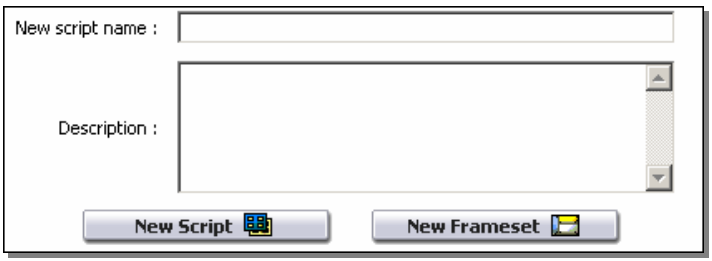
Click on the hyperlink **Click here to download the file** to open a file download popup and select a location to store the file (with extension .wss) locally.

Del.: Click on this icon to delete the selected script.

3.2 Create a new script

Create a new script:

By clicking on the command bar, an entry area appears and allows you to enter the name and a description of the new script you wish to create.



Click on to create a Pages Script.

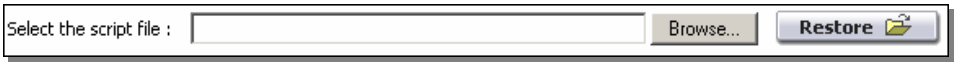
Click on to create a Frame Script.

3.3 Restore scripts

The Web Scripter™ V3 allows you to restore scripts from a file.

Restore a script from a file

By clicking on the command bar, a new field is displayed on the screen:



Select the script file: by clicking on the button, an exploration window opens and allows selecting a local file.

Once selected, the button generates a script based on the content of this file (this file can be first exported using the export function).

If a script already exists with the same name, the application adds a unique numeric identifier at the end of the new script.

The button can be used to close the application.

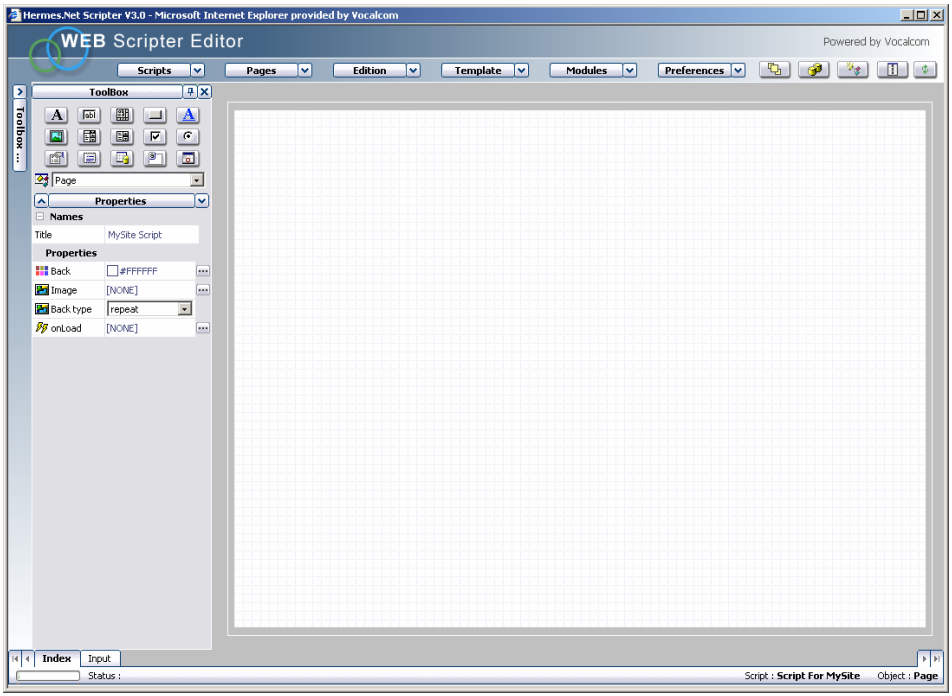
4 THE SCRIPT EDITOR

The Web Scripter application suggests working with either one of the following two scripts: Pages Scripts and « frames » Scripts. This document describes the first editor type, the « **Web Scripter Editor** » while another document provides more details on the second editor type, the « **Web Scripter FrameSet Editor** » .

This chapter contains a detailed description of the different menus of the Objects Zone and the Editing Zone.

4.1 Editor menus

The application commands have been regrouped in 2 main zones.



- The main menu (yellow-striped zone)
- The objects menu (blue-striped zone)

4.1.1 The main menu

This zone allows access to the editor's basic functions. Click on the buttons or on the arrows to open the drop-down list of the menu.

Scripts Menu

This menu gives access to the main functions of script management.

New script: Creation of a new script.

Open script: Displays the company's list of scripts. To open a script, click on 'Load' or double-click on the name of the script.

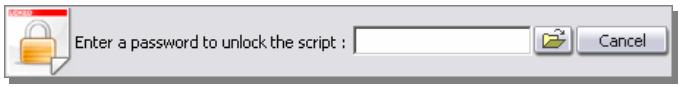
Delete script: Delete one of your company's scripts. Several scripts can be deleted in one action by holding the CTRL button for multiple selections.

Lock script: Allows blocking all access to your script (consultations and modifications) by other users. A password is required and must be confirmed on the following dialog box.

Afterwards click on Lock to confirm the protection, or on Lock and quit to confirm the protection and close the editor.

When a script is locked, a symbol is displayed on the scripts selection window, in the **Edit** column.

All the accesses to this script will need introduction of the correct password in a box at the bottom of the screen.



Generation : Allows you to test the script's content. Check out chapter « How to generate a script » on page 113.

Properties: Allows you to modify a script's properties (Name of the script, Description of the script).

Back to Home page: Lets you go back to the script management screen. (Check out chapter « Script Manager » on page 18)

Exit: Quits the application after a control of the page's status (Save or no save)

Pages Menu

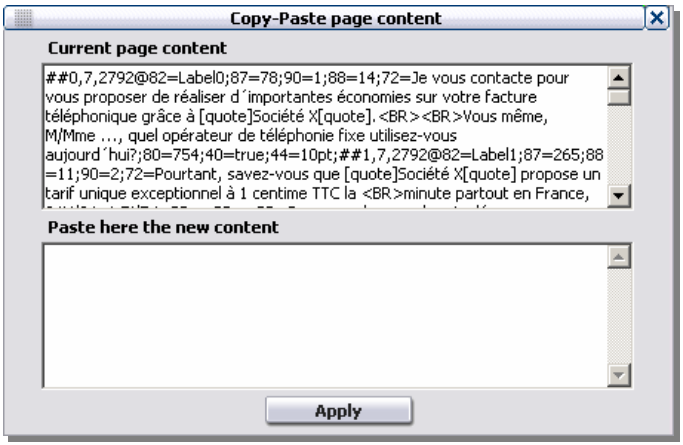
With this menu you can manage the pages that compose the script.

New Page: Creation of a new page in the active script.

Open Page: Displays the list of pages in a script and allows selecting another page to open and edit.

Reload original: Loads the page as it was at its last save. This action cancels all non-saved changes to the page.

Copy-Paste page content: This function allows you to visualize the content of the page in 'Serial' form (this format is used to transfer data between the client and the server station).


Copy-Paste page content
Current page content
##0,7,2792@82=Label0;87=78;90=1;88=14;72=Je vous contacte pour vous proposer de réaliser d´importantes économies sur votre facture téléphonique grâce à [quote]Société X[quote].

Vous même, M/Mme ..., quel opérateur de téléphonie fixe utilisez-vous aujourd´hui?;80=754;40=true;44=10pt;##1,7,2792@82=Label1;87=265;88=11;90=2;72=Pourtant, savez-vous que [quote]Société X[quote] propose un tarif unique exceptionnel à 1 centime TTC la
minute partout en France,
Paste here the new content
Apply

The zone **Current Page Content** displays the chain representing your actual page. The **Paste here the new content** zone allows you to paste another chain recovered from another script or another page. To apply the new chain to the active page, click on Apply.

« With this method you can recover the content of a certain page and send it by e-mail to another user; you can even save pages in a text file that you can store on your hard drive. »

Delete page: Allows you to select one or more pages to be deleted. Use the CTRL button for multiple selections.

Duplicate page: Create a copy of the active page; the name of the page is indexed under the running name. Ex: « Page1 » becomes « Page1 1 ».

Rename: Change the name of a page. Links using the renamed page are automatically modified.

Set as Start page: Defines the active page as the first page of the script.

Set as Search page : Defines the active page as Search page.

Share this page: By checking this box, you allow other users of your company to share a copy of your pages in their scripts through the menu 'Template > Pages'.

Print page: Gives a print preview of the page and allows you to print.

Save page as: Saves the active page under a different name.

Edition Menu

This menu allows accessing the standard edition functions.

Cut: Cut the selected object(s).

Copy: Copy the selected object(s).

Cut no connection: Cut without connection to the tables and delete the selected object(s).

Copy no connection: Copy the selected object(s) without connection to the tables.

Paste as origin: Paste the object(s) on the same X and Y coordinates as the original.

Paste: Paste the object(s) near the original object.

Paste under mouse: To paste an object under the cursor, right-click on the editing zone where you want to paste, then click 'Paste under mouse'.

Apply style: Apply the object's style to a selected object.

Apply size: Apply the object's size to a selected object.

Set as default Style : Allows defining the style (font, color,…) of the selected object as default style (it can be afterwards apply to other objects, and will be applied by default to all the new objects).

Apply style to all : Apply the default style to all the objects.

Reset default style : Restore the original default style.

Select all: Select all objects on the active page.

Unselect: Deselect all selected objects.

Group objects: see section « Object Groups » on page 45.

Ungroup objects: ungroup objects that belong to a selected group.

Set backward: Sends objects backward in a vertical plan. This function is based on the 'Z_Index' property of the objects. With each click, Z-Index=Z-Index - 1

Set forward: Brings objects forward in a vertical plan. This function is based on the 'Z_Index' property of the objects. With each click, Z-Index=Z-Index + 1

Align: These functions are only active when multiple objects are selected. With these buttons you can perfectly align the selected objects.

An object or a group of objects can be pasted on another page of the script or even in another script (only if the user change the edited script using the « Open a script » function from the « Scripts » menu, and not using the Scripts List window).

Template Menu

This menu allows access to the different templates created on the server.

Public templates / Private templates: The administrator of the Web Scripter™ platform can put pages, groups or action models at the disposal of all users. These can be particular template pages that can serve as examples. Public templates can be accessed by all users of the Web Scripter platform.

Private templates can only be accessed by users of the same company. When you authorize a page to be shared with others in the Pages menu (see section « Pages Menu » on page 27) or when you save an action as a template, all other users of your company have access to the elements. Groups created by a user are automatically shared and accessible in the template pages.

« The original templates can not be changed by the user that loads them. »

« Once loaded, pages and actions can be used as if they were created in the script themselves. »

Modules Menu

This menu gives you immediate access to the different annex applications available in the Web Scripter.

Start Upload Center

This application is used to transfer files from your client station to your private repertory on the server. This application is mainly used to transfer images from your station to the server in order to include them in your scripts. As we will explain later, the upload function allows you to download many other file types as well. For more information, check out the 'Upload Center' chapter.

Start Link builder

Use this application to create conditional chain links between pages. For more information, check out the 'Link Builder' chapter.

Start Action builder

With this application you can create actions and commands. The actions can then be linked to events on certain objects. For more information, check out the chapter on 'Action Builder'.

Start Customer File Connection

This application allows you to link your scripter objects with columns of existing tables. For more information, check out the « Connection Link », section « Connection Auto Link : » on page 144.

Start Customer File creation

With this application you can create your data files from fields on the script. For more information, check out the « Connection Link », section « Connection Create Link : » on page 146.

Start Connection Viewer

During script creation it can be useful to know the content of the tables used in a script. The 'Connection Viewer' application displays the content of the selected table and the first 10 records. For more information, check out the chapter "Connection" on page 149.

Start Connection Builder & Connection checking

With this application, you can create, modify and manage the content of your connections (tables, field types, default values…). For more information, check out the chapter "Connection Viewer" on page 149.

Disconnect all connections: Allows you to delete all connections with databases of the active script.

Replace script connections: Allows you to swap connections used in the active script.

Set global variables: Lets you define and/or initialize the global variables used in the active script. For more information, check out the chapter 'Actions and commands'

Preferences Menu

With this menu you can define the default parameters of your editor interface. All information entered in this menu is stored for each of the Web Scripter's users.

Show grid: Displays or hides the editor's positioning grids.

Grid spacing: Defines the size of the grid in pixels (Ex: 5, 10, 20, 50)

Snap to grid: Allows automatic alignments of objects on the grid.

Light Grid & Text: When using a rather dark font, it is recommended to use a grid of a lighter color for better contrast and readability.

Display object names: Display or hide object names in the editor.

Select group by click: If multiple objects are grouped, the user can choose how to select the group: If the option is checked, a simple click on one of the objects will select the whole group; if the option is unchecked, only the object itself is selected. The object group can be selected with 'ALT' + click on one of the objects in the group.

Auto Save: The Web Scripter™ offers you the possiblity to automatically save your running page every defined delay. Obviously this delay is put to zero after a save by the user. The progression of the delay is shown in the left-bottom corner.

Beware!!! If two people are authorized to work on the same script, you have to make sure that the auto-save function does not erase the modifications entered by the other person.

« The Auto Save function is very useful when a user leaves his browser open while he is not working on the script. In fact, the Auto Save function guarantees that the client session is not lost on the server »

Expert Mode: Display all object properties. If unchecked, only the basic object properties are displayed and categorized by topic.

Main Menu Buttons

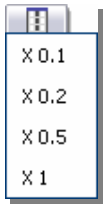
These buttons allow showing useful objects information, and also allow opening the thumbnails window.

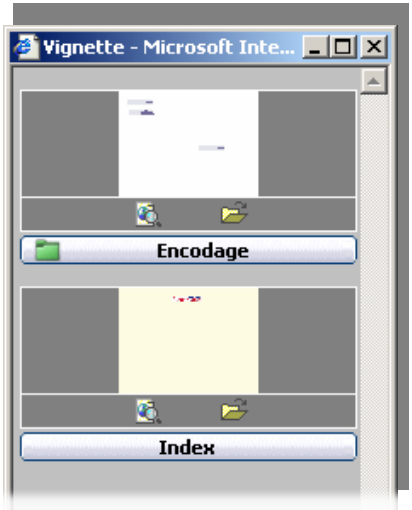
Press this button to display the **grouped objects** of the script. Let go to hide the objects.

Press this button to display the objects of the script **linked to a table.** Let go to hide the objects. This button can be used to rapidly detect objects using invalid connection (a flicking icon is visible in this case).
(see also section « Invalid connections check at script opening » on page 148 for more details on connections check.)

Press this button to display the objects of the script **linked to an action.** Let go to hide the objects. This button can be used to rapidly detect objects using invalid action (a flicking icon is visible in this case).

Click on this button to open the thumbnails window. First select the display size in the list : x 0.1, 0.2, 0.5 or 1 (= original size). The thumbnails window is displayed vertically at the left of the screen and allows to reach a page of the script by clicking on the button under the desired thumbnail. The button allows viewing the final result of the page when you put the mouse over it.

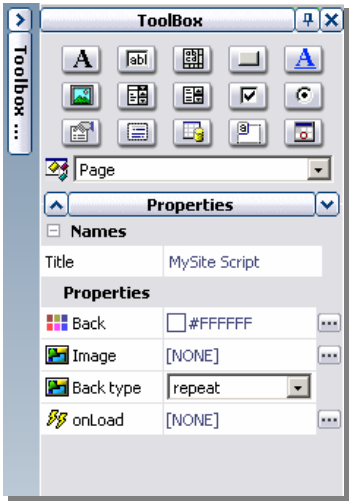




Click on this button to display the editor again. This action will very rarely be used.

4.1.2 The Objects Menu

The object zone is on the left side of the screen. It consists of an object toolbar, a script content navigation zone and an object property display zone.



By default, this zone is hidden, and it can be shown using the vertical **Toolbox** button.

By clicking on it, the zone is deployed and stays opened until the mouse is used on one of its buttons or fields.

If you want to leave this zone deployed, click on the button at the top of the zone.

The toolbar

The toolbar allows you to add different kinds of objects on the active page.

To add an object to your page, simply click on the one you wish to add.

- **LABEL**: Create a text area on the page.
- **TEXT BOX**: Create a text box.
- **TEXT AREA**: Create a text area.
- **BUTTON**: Create a command button.
- **HYPERLINK**: Define a HREF hyperlink.
- **IMAGE**: Create an image area.
- **DROPDOWNLIST**: Create a drop-down list.
- **LIST BOX**: Create a multiple-selection list.
- **CHECK BOX**: Create a checkbox.
- **RADIO BUTTON**: Create a multiple-option button.
- **OBJECTIONS**: Create an objection list.
- **IFRAME**: Insert a page or URL in the active page.
- **DATAGRID**: Create a data table.
- **LITERAL**: Create a HTML content area.
- **CALENDAR**: Insert the Web Agenda applet.

The Content Zone

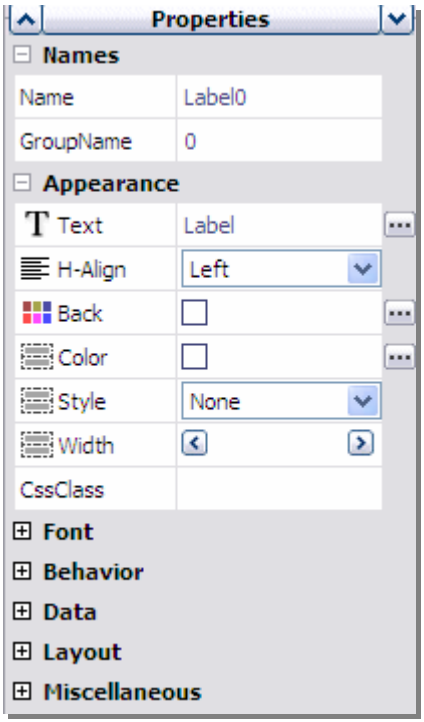
This zone gives you direct access to a page or an object on the active page.

The Properties Zone

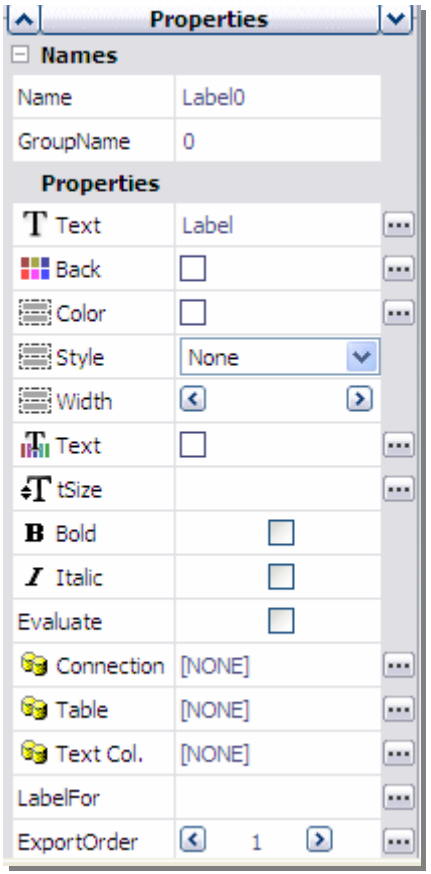
This zone looks different when the « Expert Mode » option is checked than when it is not.

Expert Mode checked

Expert Mode unchecked



If the expert mode is checked, the properties are grouped by topic. The available topics depend on the selected object. To open the properties of a certain topic, click on the small '+' icon.

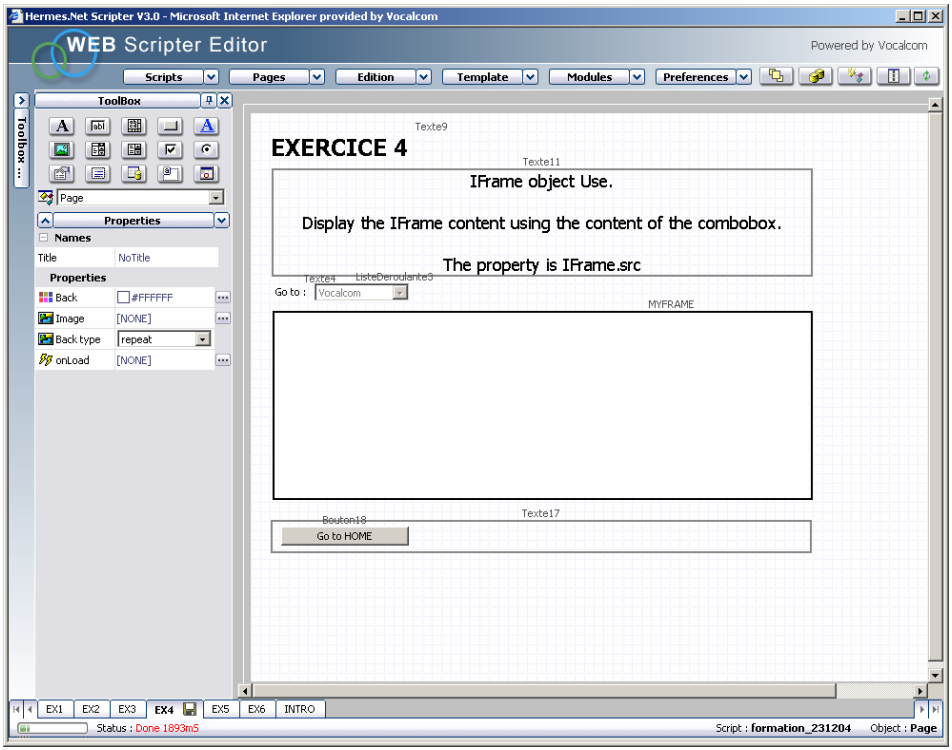


When the expert mode is not active, the object's main properties are displayed.

Certain properties require a configuration pop-up or suggest help in entering the information; to see the data entry pop-up, click on .

The different elements displayed in this zone are described in the next chapter, in the section « Property Overview » on page 56.

4.2 The Edition Zone

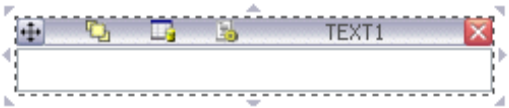


This zone (yellow-striped) is in the middle of the screen and displays the script page as it would look during execution.

All objects are visible and can be selected by clicking on them. The properties visible on the left side and the main menu are adapted in function of the style of the chosen object.

4.2.1 Select an Object

When an object is selected, its appearance changes in order to provide the user with the following information and functionalities:



Move object

Make object part of a group

Link object to a table

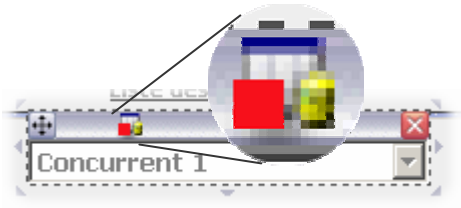
Let object launch an action

Give object a name

Delete object

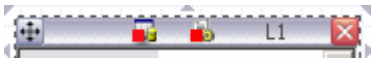
Resize object

When an object linked to the DB is selected, and the connection is not available, a flicking red square is displayed in the properties of the object. (see next picture)



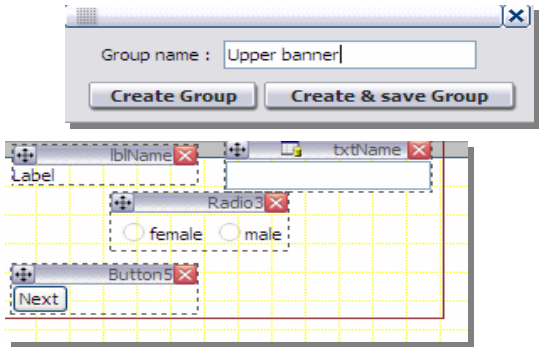
See also section « Invalid connections check at script opening » on page 148 to get information about invalid connections.

The same way when an object linked to an invalid action (for example missing after an importation), a flicking red square appears in the object properties.



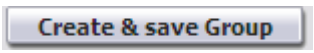
4.2.2 Object Groups

This is a time-saving option if you want to execute actions on a group of objects on the same page. You can execute the action once for all objects that belong to the group, rather than for each object individually.



If you want to group objects, there are two ways to process.

You can create a group only on the active page, for instance if you want to move objects together. Simply click on **Create a group** in the **Edition** menu, then after entering the name of the group.

You can create a group on the active page and save it in your templates. To do so, click on  on the dialog box. Your object group will be available for all users of your company.

Note on the Edition Zone

The screen zone (zone displayed with the grid) represents the size of the window in which the script will be launched. The zone does not limit the placement of objects; one can put objects outside of the screen zone. These objects are treated similarly as the others.

4.2.3 Help dialog windows

Several help dialog windows can be displayed during edition of a script in order to get syntax information to build the next expressions:

- Custom Calcul expressions on the Action Builder screen (category « Calcul »). (see chapter « Actions and Commands » on page 152)
- User functions defined on the Action Builder screen (category « User Function »). (see chapter « Actions and Commands » on page 152)
- Custom formats (see property **Format** of objects containing text)

The 2 first dialog windows are reached using the button

at the bottom of the Action Builder screen, the 3rd one is accessible via the button in the title bar of the dialog box « Select a Format » related to the **Format** property.

Custom Calculs help screen

Type	Designation	Symbol	Example
Mathematic Operators	Addition	+	
	Subtraction	-	
	Multiplication	*	
	Division	/	
	Modulo	%	
Increase Operators	Increase	++	i++ add 1 to i
	Decrease	--	i-- subtract 1 to i
Assignation Operators	Affectation	=	
	Addition and assignation	+=	x+=2 equals to x=x+2

	Subtraction and assignation	-=	
	Multiplication and assignation	*=	
	Division and assignation	/=	
	Modulo and assignation	%=	
	Binary AND and assignation	&=	
	Binary OR and assignation	|=	
	Binary exclusive OR and assignation	^=	
Comparison Operators	equal	==	1==1 is TRUE
	not equal	!=	1!=1 is FALSE
	strictly equal	===	
	strictly not equal	!===	
	less	<	
	more	>	
	less or equal	<=	
	more or equal	>=	
Bit to bit Operators	Binary AND	&	
	Binary OR	|	
	Binary exclusive OR	^	
	Bit complement	~	
	left bit offset	<<	

	right bit offset	>>	
Logical Operators	NOT	!	
	AND	&&	
	OR	||	
Mathematical functions	absolute value	Math.abs(value)	Return the absolute value of **value**
	Maximum	Math.max(value1,value2)	Return the max value between **value1** and **value2**
	Minimum	Math.min(value1,value2)	Return the min value between **value1** and **value2**
	Random	Math.random()	Return a real between 0 and 1
	Round	Math.round(value)	Return the round of **value**
Conversions functions	String > Integer	parseInt(string)	
	String > Float	parseFloat(string)	

User Functions help screen

Type	Examples / Explanations

Conditional branch	**if** (condition) action;
	if (condition)
	{
	action1;
	action2
	}
	if (condition)
	{
	action1;
	action2
	}
	else
	{
	action3;
	}

Loops	**while** (condition)
	{
	action1;
	action2
	}
	for (start value ; end value; step)
	{
	action1;
	action2
	}
	
	Example :

	for (i=1 ; i<=20 ; i++) {sum+=i;}
String manipulations	1 – Look for a character MyString.charAt(num) Return the char **'num'** of the string **MyString** 2 – Look for the position of a string in another string MyString.indexOf(SearchString,[start]) Return the position of the first char of **SearchString** into **MyString**, as from the position **start**. Return -1 if not found. 3 - Idem 2 beginning by the end of the string MyString.lastIndexOf(SearchString,[start]) Return the position of the first char of **SearchString** into **MyString**, beginning by the end of the string. Return -1 if not found. 4 – Return a string MyString.substring(indexA, indexB) Return char found between position **indexA** and position **indexB**

Conversion	Uppercase > Lowercase MyString.toLowerCase() Lowercase > Uppercase MyString.toUpperCase()

Custom Formats help screen

Example	Regular Expression	Meaning
Digits only	^\d*$	**^** : mean the begin of line
		\d : mean a digit
		***** : 0 to n events of the preceding character
		$: mean the end of line
Characters only	^[a-zA-Z]*$	**^** : mean the begin of line
		[a-zA-Z] : a no case sensitive character
		***** : 0 to n events of the preceding character
		$: mean the end of line
Begin with 0, 1 or 2	^[0-2]+	**[0-2]** : a digit between 0 and 2
		+ : any character
Contient le mot "hello"	hello .*|^hello$	**bonjour .*** : the "hello" word and a blank space
		| : "or"
		^bonjour$: only the "hello" word
Ten digits only	^\d{10}$	**{n}** : n events of the preceding character
Begin with one digit and finish with one	^\d.*.[a-z]$	**^\d** : begin with a digit
		[a-z]$: ended with a lower-case

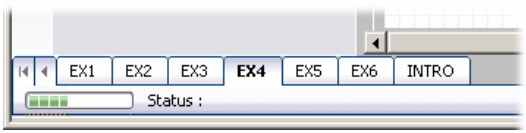
character		character **.*.** : any alpha numeric character between the digit and the character

Extra information are also available on an external web site accessible via the button

.

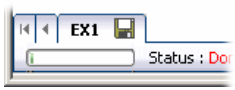
4.3 Navigation Tabs

At the bottom of the Web Scripter window, tabs are displayed. They allow reaching directly a page of the script by clicking on its name.

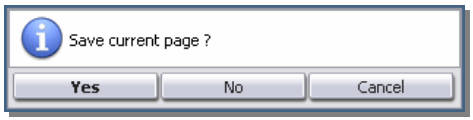


In this example, the script is composed of 7 pages.

When modifications have been done to the current page, a symbol is displayed next to the modified page: it allows knowing directly the saving state of the page.



If you click on another page tab, a dialog box asks if you want first to save (or cancel) the last modifications before the next page is displayed.



If you click **Yes**, last modifications are saved before the new page is displayed.

If you click **No**, last modifications are lost.

If you click **Cancel**, this dialog box is closed and you stay on the current page.

5 OBJECTS AND PROPERTIES

This chapter explains all properties of the objects and the page used in the editing standard.

5.1 Property Overview

Different object types have many properties in common. Please find below a description of those common properties.

Expert: ☑ Only accessible in Expert Mode ☐ Accessible in all modes

Property	Expert	Description	Objects involved	Property format
NAMES				
Name	☐	When you add an object to the editor, the application automatically creates a name for that object. The name appears in the property: Names > Name. The user can change the name, but there are some restrictions on the (number of) characters: Only alpha-numerical characters and '_' (underscore) are accepted. The name must have at least 2 characters. All throughout the editor, this name will refer to the object in question. When you create a customer file from a script, it is this name that will be used by default to name	All, except page	Text

		the customer table column.		
GroupName	☐	If an object belongs to a group, the property contains the name of the group. To change a group name, select all objects of the group and enter a new name. To ungroup an object, delete its group property by deleting the name of the group.	All, except page	Text
Title	☐	This property gives a title to the page of the browser.	Page	Text

APPEARANCE				
AlternateText	☑	Text that appears when the mouse points over the image and instead of the image before it is displayed.	Images	Text
Text	☐	Allows you to enter the text to be displayed in the object. For a 'text' object, the property contains the HTML code to be included in the text area.	Label, Textbox, TextArea, Href Link, Button, Literal.	Text
H-Align	☑	ALlows you to choose the text alignment inside the object.	Label, TextBox, TextArea, Button, DataGrid, CheckBox, RadioBox, Objection.	List of options
Back	☐	Object fill color.	Label, Texbox, TextArea, Button,	Color Code*, or Color PopUp

			HrefLink, Image, Combo, ListBox, CheckBox, RadioBox, Objection.	
Color	☐	Object line color.	Label, Texbox, TextArea, Button, Href Link, Image, CheckBox, RadioBox, Objection, DataGrid.	Color Code*, or Color PopUp
Style	☐	Line style.		
Width	☐	Line width.		
Image	☐	Image to be displayed in the background of the DataGrid or the page	DataGrid, Image, page	List of options
Back Type	☐	Defines how the background image will appear on the page (repeated image or not).	Page	
Frame Border	☑	Iframe border visible or not.	Iframe.	CheckBox YES/NO
CSS Class	☑	Selection of the style type present in the CSS File. (Check out the section Upload Center to find out how to upload a Css File).	All except Literal, Agenda and page	Text
LineHeight	☑	Height of the text displayed in the field in nb of lines	Label	nb of lines
BgColor	☐	Background color of the table borders.	DataGrid.	Color Code*, or PopUp Colors
Alternate	☑	Background color of the alternated lines.		

Selected	☐	Background color of the alternated lines		
Over	☑	Line color when cursor is placed on the line.		
Header	☐	Background color of the table header.		
Selected	☐	Text color of the selected line.		
Header	☐	Text color of the title.		
GridStyle	☑	Class style assigned to the datagrid		Text

* Color Code: #FFFFFF (Three Hexa values representing red, green and blue)

FONT				
FontName	☑	Select the object's text font.	All except Images, Iframe, Literal.	Text or PopUp font.
Text	☐	Select the text color.		Color Text*, or PopUp Colors.
tSize	☑	Select the text size.		Size in Points (1pt, 5pt, ..), or Pixels, or PopUp font size
Bold	☐	Bold text.		CheckBox YES/NO
Italic	☑	Text in italics.		
Overline	☑	Text with a line above.		
StrikeOut	☑	Strike out text.		

U Underline	☑	Underline text.		
B Selected	☐	Selected Datagrid line in bold.	Datagrid	CheckBox YES/NO

BEHAVIOR				
Allow Null	☑	The data entry field can be left blank.	Textbox, TextArea.	CheckBox YES/NO
Format	☐	Allow to define a mask to type text. Opens the Popup format. Check out the PopUp Format chapter.		PopUp Format.
Alert	☑	Defines the alert message to be displayed when the format is not respected.		Text.
AccessKey	☑	Define which button allows you to access the object.	All except DataGrid, IFrame, Literal.	List of options.
TextMode	☑	Define if the line must be displayed as a password zone.	TextBox	CheckBox YES/NO
TabIndex	☑	Define the order of the object's tabulation. Enter a number between 1 and nb Objects.	All except Label, Href Link, Image, DataGrid, IFrame, Literal.	Number or PopUp Tabulation.
Selection Mode	☑	Define whether multiple items can be selected from a list. Multiple selections are done by click + CTRL.	ListBox	List of options.

Visible	☑	Displays or hides the objects	All except IFrame	CheckBox YES/NO
Evaluate	☐	This property is explained in more detail in the chapters about objects. It allows the application (in preview mode) to verify if the text zone (text property) contains a global variable ($) or a form field (@). If yes, the variable will be replaced by its content.	Label, TextBox, TextArea	CheckBox YES/NO
ReadOnly	☑	If this property is checked, the field can only be displayed, not modified.	Texbox, TextArea	CheckBox YES/NO
ShowHeader	☑	Allows you to display the table heading.	DataGrid	CheckBox YES/NO
PageSize	☐	Define the maximum number of lines per page to be displayed in the table.		Number.
Allow Sort	☐	Indicates if the table content can be sorted.		CheckBox YES/NO
ToolTip	☑	Displays the text when the cursor is placed on the object.	All except DataGrid, IFrame, Literal, page	Text.
Enabled	☑	Choose if an object is active or not.	All except Image, Iframe, Objections, Literal.	CheckBox YES/NO
Scrolling	☑	Displays the scroll bars of the Iframe object.	Iframe.	List of options.

		Auto: the scroll bars are displayed in function of the content. No: the scroll bars are not displayed. Yes: the scroll bars are always displayed.		
DisplayType	☐	Allows you to define the default agenda settings: Month, week, and day.	Agenda	List of options.
Planning	☐	Allows you to choose on which agenda campaign the script can be opened		
Commercial	☑	Allows assigning a commercial to the agenda at startup		
Subject	☑	Defines agenda subject		
Note	☑	Defines agenda default note		
Test Format	☑	Prohibit test format of page fields.	Button, Image, Href Link.	CheckBox YES/NO
SaveForm	☐	Prohibit saving page fields in the database.		

LAYOUT				
Left	☑	Defines the horizontal position of the left side of the object.	All	Number.
Top	☑	Defines the vertical position of the top border of the object.		Number.
Width	☑	Defines the object's width.		Number.
Height	☑	Defines the object's height.		Number.

Header	☑	Defines the height of the table header.	DataGrid.	Number.
Line	☑	Defines the height of the table line.		Number.
LeftMargin	☑	Defines the document's left margin.	Page.	Number.
TopMargin	☑	Defines the document's top margin.		Number.
Z-Index	☑	Defines the object's Z position. Each object is positioned on a plan. This property defines on which plan the object is placed.	All.	Number.
CellSpacing	☑	Defines the space between each cell of the table.	DataGrid.	Number.
Cell Padding	☑	Defines the internal space of a table cell.		
Border Size	☑	Defines the width of the cell borders.		
Wrap	☑	Defines if the text displayed or entered in the object should fit the window size.	TextArea	CheckBox YES/NO
MarginWidth	☑	Defines the object's left internal margin.	Iframe.	Number.
MarginHeight	☑	Defines the object's top internal margin.		

DATA				
Connection	☐	Defines the connection used to populate the object.	All except Href Link, Button,	List of options.

			Image, Iframe, Literal.	
Table	☐	Defines the table to be used for this object.	All except Href Link, Button, Image, DataGrid, Iframe, Literal.	List of options.
Text Col.	☐	Defines the column in which the object's content should be saved.		List of options.
Value Col.	☐	Defines the column in which to import the object's value.	Combo, ListBox, CheckBox, RadioBox.	List of options.
Request	☐	Defines the request to be executed to populate the object.	DataGrid.	Text.
LastName, FirstName, Adr1,...	☑	Defines values of fields included in the Agenda object (see section « The CALENDAR Object » on page 109 for the complete list.	Agenda	Text.

ACTIONS				
OnLoad	☐	Action when loading a page.	Page.	PopUp Actions.
OnClick	☐	Action when clicking on an object.	TextBox, TexArea, Button, Href Link, Image, CheckBox, RadioButton.	PopUp Actions.
OnChange	☑	Action when changing an object's content.	TextBox, TextArea, Combo, ListBox.	PopUp Actions.

OnSelect	☑	Action when selecting a component of an object.	TextBox, TextArea, DataGrid.	PopUp Actions.
OnDbl. Click	☑	Action when double-clicking an object.	Button.	PopUp Actions.
On MouseOut	☑	Action when the cursor moves outside the object zone.	Button, Href Link, Image.	PopUp Actions.
On MouseOver	☑	Action when the cursor is placed over the object.	Button, Href Link, Image.	PopUp Actions.
On KeyPress	☑	Action when pressing a keybord button.	TextBox, TextArea.	PopUp Actions.
Apoint. Add.	☑	Action when an appointment is added into the agenda	Agenda	PopUp Actions.
Apoint. Upd.	☑	Action when an appointment is updated into the agenda		
Apoint. Del.	☑	Action when an appointment is deleted		

MISCELLANEOUS				
Dynamic Contents	☑	Allows filling the object's content with a request to a database. For more information check out the Objection object.	Objection.	PopUp Dyn Objections.
Dynamic List	☑	Allows filling combo or drop-down list content with a request to the database. For more information, check out the Combo and Drop-down list objects.	Combo, ListBox.	PopUp Dyn List.

Objection List	☐	Allows filling the objection list.	Objection.	PopUp Objections
Static Items	☐	Defines the static content of objects.	Combo, ListBox, CheckBox, RadioButton.	PopUp Static List.
Link	☐	Defines the destination page when clicking on objects.	Button, Href Link, Image.	PopUp Link
External URL	☐	Allows referring the script to an external URL.		Text.
URL	☐	Configures the URL in the object.	Iframe.	Text.
Target	☐	If an external URL is configured, this property allows you to define or display that URL.	Button, Href Link, Image.	List of options.
Columns	☐	Defines the number of characters displayed per line. This property inhibits the object's size properties.	TextBox, TextArea.	Number.
Rows	☐	Defines the number of lines displayed by the object. This property inhibits the object's height properties.	TextArea, ListBox.	Number.
Labelfor	☐	This property allows you to link a label with a data entry object (TextBox, TextArea, etc....)	Label.	Name of an object of the page.
Change Planning	☑	Indocates if the agent can change the planning assigned to the agenda	Agenda	CheckBox YES/NO
Add Appointm.	☑	Indicates if the agent cana dd appointments		

Move Appointm.	☑	Indicates if the agent can move appointments		
Edit Appointm.	☑	Indicates if the agent cane dit appointments		
Del. Appointm.	☑	Indicates if the agent can delete appointments		

5.2 The Objects

5.2.1 PAGE Object

As all editor objects, the page itself has its own properties.

Description of Properties

Name Category

Title: The title appears at the top of the browser window.

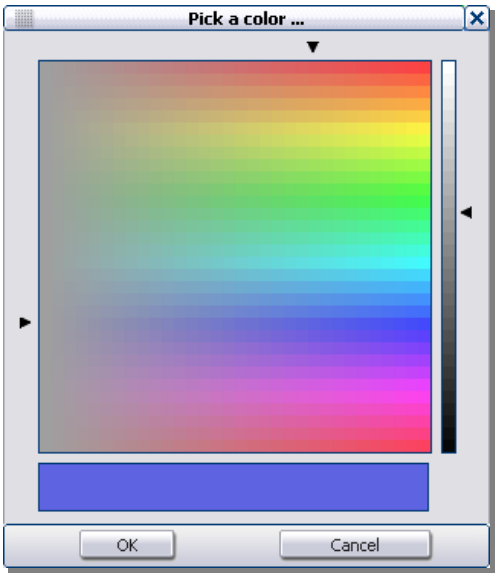
Example: Title = My Test Page

Appearance Category

Back: Choose your page's background color

Example: Background = #FF0000 > the page background will be red.

Click on the ... button to open the **Pick a color** window :

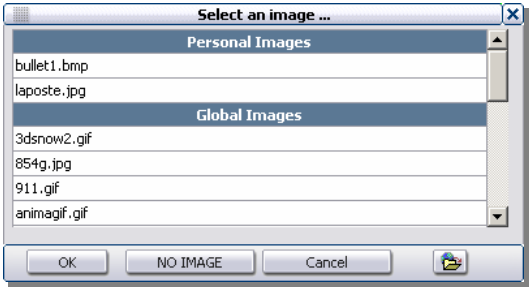


The zone at the bottom indicates the current color.

The vertical ruler can be used to select the luminosity level of the whole palette from white to black. Just click on the desired level and you will see that the whole palette is adapted. Just click on the desired color within the palette and click on the OK button to confirm your choice. Click on the Cancel button to cancel and restore the previous color.

Image: Choose a background image for your page.

To select the background image, click on the ... icon and choose from the images available.



Two types of images are available: personal and global images. Global images are part of the scripter; personal images belong to your company and can be added with the 'Upload file' module (see chapter "Upload Center" on page 135). This module can also be reached using the button at the bottom of the dialog box.

To delete an image from the background, simply open the image popup and choose .

Back Type: Choose the display type of the background image. **Repeat**: the image is repeated on the whole page; **No-repeat**: the image is placed on the top left side of the page.

Layout Category

Width: Defines the width of the page (in pixels).

Height: Defines the height of the page (in pixels).

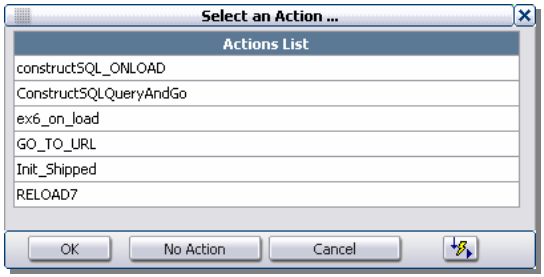
LeftMargin: Defines the page's left margin (in pixels).

TopMargin: Defines the page's top margin (in pixels).

Action Category

onLoad: Allows you to choose an action to be executed when loading a page.

Click on the ... button to open the dialog box **Select an Action**.



The list of existing actions is displayed, you have to select the right action and to assigne it to the action using the OK button. If you want create a new action, launch the **Action Builder** module using the button. See chapter “Actions and Commands” on page 152 to get details about actions and commands creation.

Tips and tricks

Change the background color of a page to red via a custom action: document.body.bgColor = "#FF0000" ;

Change the title of the active page via a custom action: document.title = "My test page" ;

5.2.2 LABEL Object

With this object, you can display text on the screen in different formats.

Description of Properties

Name Category

Name: This property allows you to name the object. Each object of a page must have a unique name. If the name is not unique, the editor will display a warning.

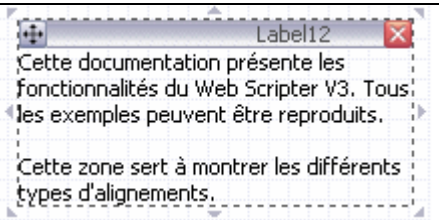
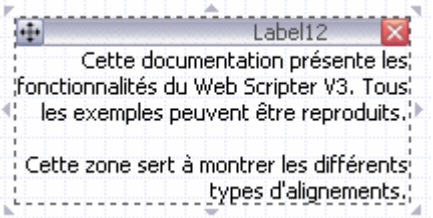
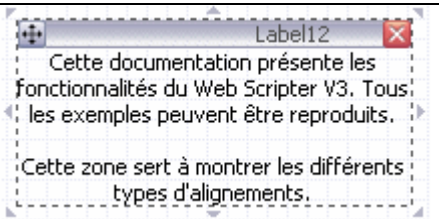
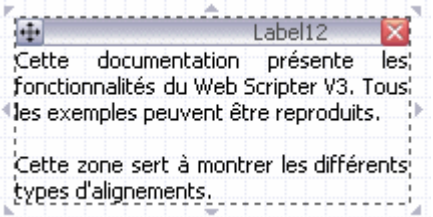
GroupName: This property represents the name of the group to which the object belongs. For more information on groups, check out the chapter « Object Groups » on page 45.

Appearance Category

Text: Enter text directly in the zone or click on to display the data entry popup and enter your text. Carriage return characters are accepted in the popup Text. If carriage return characters are entered in the popup text, they are replaced by the '<BR>' mark in the text zone of the property. The same way other HTML tags will be replaced, for example to put text in bold characters.

H-Align: Choose the text alignment type in the object zone.

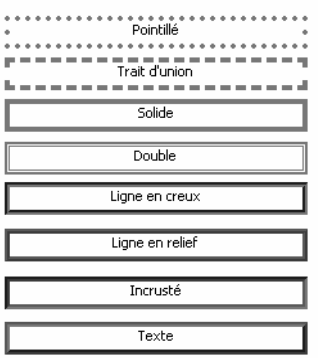
Different types of alignment

Left	
Right	
Center	
Justify	

Back: See « PAGE Object »

Color: Choose the object's border color.

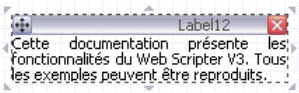
Style: Choose the object's border style.



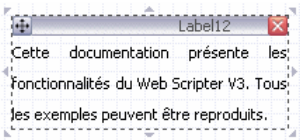
Width: Choose the width of the object's border.

CSS Class: Choose the style to apply to the object. The style must belong to the styles loaded in the 'Upload Center' module.

LineHeight : Specify here the height of each text line of this field, in number of lines.


Label12
Cette documentation présente les fonctionnalités du Web Scripter V3. Tous les exemples peuvent être reproduits.

Example **LineHeight** = 1


Label12
Cette documentation présente les fonctionnalités du Web Scripter V3. Tous les exemples peuvent être reproduits.

Example **LineHeight** = 2

Font Category

FontName: Choose the font type in which to display the label's content.

Text: Define the color of the text to be displayed.

tSize: Define the text size.

Bold: Indicate whether the text should be displayed in bold.

Italic: Indicate whether the text should be displayed in italics.

Overline: Indicate whether there should be a line above the text.

StrikeOut: Indicate if the text should be struck out.

Underline: Indicate whether the text should be underlined.

Behavior Category

AccessKey: Assign an access key to the object. If the 'AccessKey' property has the value 'E', the operator can access the object by pushing simultaneously on 'ALT' + 'E'.

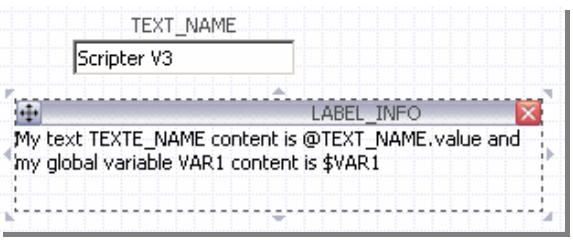
Visible: Display or hide the object.

Evaluate: If this box is checked, the 'Text' content of the object is evaluated before being displayed, meaning that the 'Text' property of the object can contain global variables or names of the page fields.

Important: Global variables of the script are referenced with a **$** and page objects with a **@.**

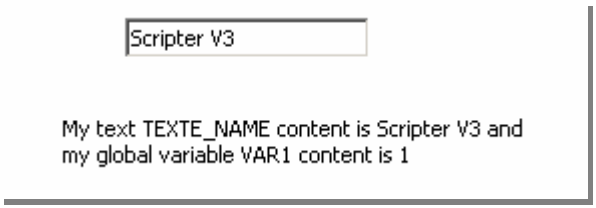
Example

In this example, a TEXTBOX object with the name « TEXT_NAME » has been populated with the text « ScripterR V3 »; its content can be used in the script thanks to the reference « @TEXT_NAME.value ». A global variable $VAR1 was created and initialized to the value "1".



Another text object carrying the name « LABEL_INFO » is populated with the text above.

If the property 'Evaluate ' is checked on the object LABEL_INFO, the two values @TEXT_NAME.value and $VAR1 will be replaced when the page is displayed.
When the page is generated, the result will be:



Tooltip: Define the text displayed in the object's Tooltip.
Enabled: Allows you to inhibit the object. The object's functions (click, Load, Color, etc.…) are ignored.

Data Category

In this category, you can define how to connect objects to your databases. We will see further on in this document that it is possible to automatically link all script fields to an existing table or even to automatically create a table according to script objects.

Connection: Defines which connection must be used.
Table: Defines the table in which the chosen column resides.
Text Col.: Defines the column of which the content must be displayed in the object.

Click on the button to open the dialog box **Select a Connection**.

The existing connections are proposed in the list **DataConnection**. Once selected, tables detected within this database are proposed in the lists **DataTable** and **DataTextField**.

The button can be used to open the **Manage Connections** screen. See chapter “Connection Viewer” on page 149.

Layout Category

Left: Choose the position of the object’s left border (in pixels).

Top: Choose the position of the object’s top border (in pixels).

Width: Choose the object’s width (in pixels).

Height: Choose the object’s height (in pixels).

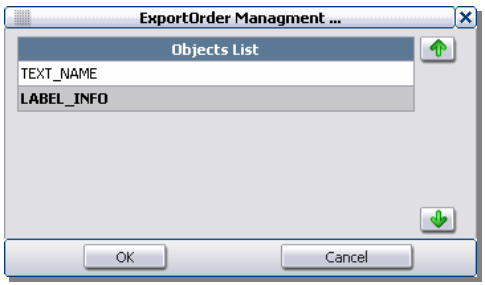
Z-Index: Choose the object’s plan. An object with a low Z-Index value will be recovered by an object with a higher Z-Index.

Miscellaneous Category

LabelFor: Assign a label to a data entry field.

If the property **LabelFor** of the NOM object is linked to ‘CHAMP_NOM’, a click on the text ‘Nom :’ will access the field ‘CHAMP_NOM’.

ExportOrder: allows you to place objects in the generated page in a specific order. One can directly indicate the order or click on the button to open the window **ExportOrder Management** to define the order. All fields are presented in the list in the export order, and the and buttons allow you to move the selected object up or down the list.



Tips and tricks

Get or change the content of a Label object:

Example of JavaScript code that can be used into a custom command :

If the Label object is named ‘My_Label’

$My_Variable = @My_Label.innerHTML ;

@My_Label.innerHTML = $My_Variable ;

@My_Label.innerHTML = ‘Hi, my name is …’

Use the Label object to create text zones:

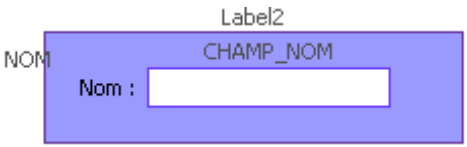
Add a label object

Enter a blank text in the 'Text' property (enter a space).

Resize the area.

Choose a background color.

Give it a solid 1px border.



Enter a Z-Index property of 1

Then place the objects in the new zone.

Do not forget that the objects placed in this zone must have a Z-Index higher than 1.

5.2.3 TEXT Object (TEXTBOX)

With this object, you can display and enter text.

Description of Properties

Name Category

See object « LABEL »

Appearance Category

Text : See object « LABEL »

H-Align : See object « LABEL »

Back : See object « LABEL »

Color : See object « LABEL »

Style : See object « LABEL »

Width : See object « LABEL »

CssClass : See object « LABEL »

Font Category

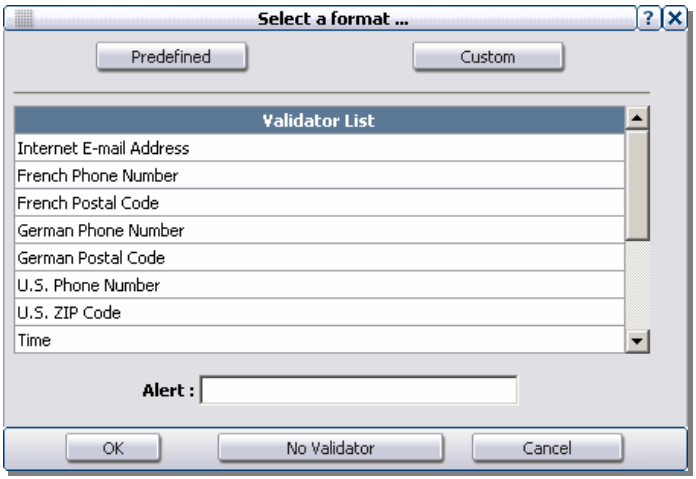
See object « LABEL »

Behavior Category

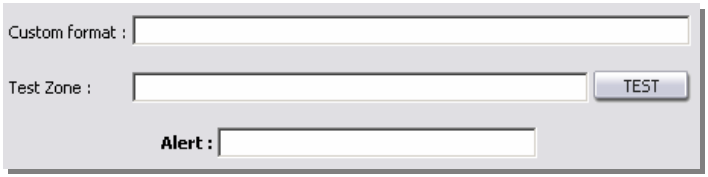
Allow Null: This property allows you to define if an agent can leave a field empty. If the page is authorized with a blank field, the message specified in the 'Alert' property will be displayed with an exclamation mark next to the object in question.

Format: With this property, you can specify an entry format for the object. For more information on formats, check out chapter « Text format via cascading style sheet ».

If you click on the button , you will open the **Select a format** window :



This window allows assigning a predefined format (e-mail address, phone numer, zip code,…) to the current field. You can also create a new format using the Custom button.



Enter a **Custom format** based on the regular expressions. (see section “Help dialog windows” on page 47). Use the field **Test Zone** and the TEST button to check if your custom format is correct.

The **Alert** field can be used to indicate the error message to be shown when you click on the flicking symbol ! close to the erroneous field.

If the page is validated and the text format does not correspond to the format defined, the message in the ‘Error’ property will display an exclamation mark next to the object in question.

Alert: This property allows you to enter the error message that will appear when the specified format for the field is not respected.

AccessKey: See object « LABEL »

TextMode: Choose the display type of the text zone: **SingleLine** or **password**. In function of the browsers, the characters of a field with a password are replaced by the reference mark * or bullets.

TabIndex: See object « LABEL »

Visible: See object « LABEL »

Evaluate: See object « LABEL »

ReadOnly: Prevent entries in the field.

ToolTip: See object « LABEL »

Enabled: See object « LABEL »

Data Category

See object « LABEL »

Layout Category

See object « LABEL »

Action Category

onClick: Link an action to the event 'Click' on the object.

onChange: Link an action to the event 'Modify' on the object. (If the text is modified)

onSelect: Link an action to the event 'Selection' on the object. (If part of the text is selected)

onKeyPress: Link an action to the event 'press' on the object. (If a button is pressed in the text zone)

See property "**onLoad**" of the **Page** object on page 70 to get details about the window **Select an action**.

Miscellaneous Category

Columns: Replace the property 'Width' (in pixels) of the textbox object by a width in number of characters.

ExportOrder: See object « LABEL »

Tips and Tricks

Recover or change the content of a TextBox:

If the object TextBox is named ‘My_Text’

$My_Variable = @My_ Text.value ;

@My_Text.value = $My_Variable ;

@My_ Text.value = ‘Hi, my name is …’

5.2.4 TEXT Zone Object (TEXTAREA)

With this object, you can display and enter text in a multiple-line area.

Description of Properties

Name Category

See object « LABEL »

Appearance Category

See object « TEXT Object (TEXTBOX) »

Font Category

See object « LABEL »

Behavior Category

Allow Null : See object « TEXT Object (TEXTBOX) »

Format : See object « TEXT Object (TEXTBOX) »

Alert : See object « TEXT Object (TEXTBOX) »

AccessKey : See object « LABEL »

TabIndex : See object « TEXT Object (TEXTBOX) »

Visible : See object « LABEL »

Evaluate : See object « LABEL »

ReadOnly : See object « LABEL »

ToolTip : See object « LABEL »

Enabled : See object « LABEL »

Data Category

See object « LABEL »

Layout Category

See object « LABEL »

Wrap: Indicate if the text should automatically go to the next line if larger than the field size.

Action Category

onChange: See object « TEXT Object (TEXTBOX) »

onSelect: See object « TEXT Object (TEXTBOX) »

onKeyPress: See object « TEXT Object (TEXTBOX) »

See property "**onLoad**" of the **Page** object on page 70 to get details about the window **Select an action**.

Miscellaneous Category

Columns: See object « TEXT Object (TEXTBOX) »

Rows: Replace the 'Height' property (in pixels) of the text area object by a height in number of lines.

ExportOrder: See object « LABEL »

Tips and Tricks

Recover or modify the content of a Textarea object:

If the TextArea object is named 'My_Text'

$My_Variable = @My_ Text.value ;

@My_Text.value = $My_Variable ;

@My_ Text.value = 'Hi, my name is …'

5.2.5 BUTTON Object (BUTTON)

This object allows you to launch actions such as going from one page to another, displaying a message, or launching an action.

Description of Properties

Name Category

See object « LABEL »

Appearance Category

See object « TEXT Object (TEXTBOX) »

Font Category

See object « LABEL »

Behavior Category

AccessKey: See object « LABEL »

TabIndex: See object « TEXT Object (TEXTBOX) »

Visible: See object « LABEL »

Evaluate: See object « LABEL »

ToolTip: See object « LABEL »

Enabled: See object « LABEL »

Test Format: Define if the format of the data entered on the page must be tested or not. If the box is unchecked, the conditions encoded in the property 'Behavior > Format' of each field will not be tested.

Save Form: Specify if the data entered on the page must be saved by clicking on the button.

Layout Category

See object « LABEL »

Action Category

onClick: See object « TEXT Object (TEXTBOX) »

onDblClick: Link an action to the event 'Double-Click' on the object.

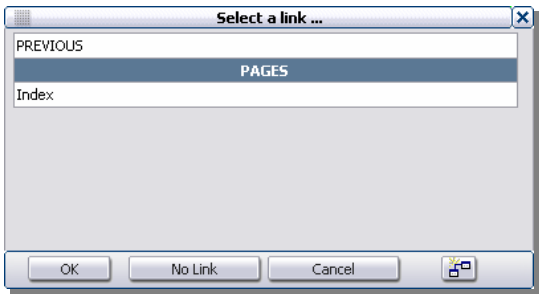
OnMouseOut: Link an action to the event 'Cursor outside of object zone'.

OnMouseOver: Link an action to the event 'Cursor over object zone'.

See property "**onLoad**" of the **Page** object on page 70 to get details about the window **Select an action**.

Miscellaneous Category

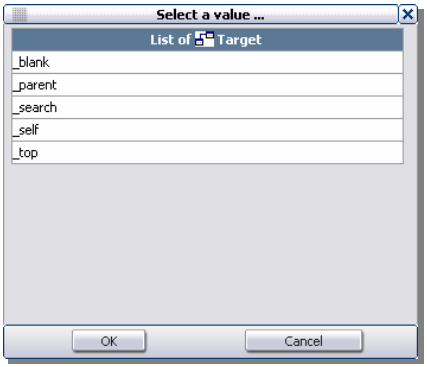
Link: Specify a destination page when clicking on the object. By clicking on the button, a window pops up and suggests the following destinations: under **PAGES**, a page of the script (Index), the previous page (PREVIOUS), or with the button, a specific link created with the 'Link Builder' application.



For more information on creating specific links, check out the chapter « Link Builder ».

External URL: Specify an Internet address to be loaded when clicking on the object. Click on the button to open the input dialog box.

Target: If the property External Url is used, select the place where you want to display the chosen Internet pages.



_Blank: Opens the page in a new IE window.
_Parent: Opens the page in the IE window that called for the Script application.
_Search: Opens the page in the IE search window.
_Self: Opens the page in the actual window.
_Top: Opens the page in the main IE window.

ExportOrder: See object « LABEL »

Tips and Tricks

Recover or modify the text displayed on a button:
If the object Button is named 'My_Button'
$My_Variable = @My_ Button.value ;
@My_Button.value = $My_Variable ;
@My_ Button.value = 'Hi, my name is …'

5.2.6 LINK Object (HYPERLINK)

With this object, you can display a text that serves as hypertext link.

Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

See object « TEXT Object (TEXTBOX) »

Font Category

See object « LABEL »

Behavior Category

See object “LABEL Object”

Test Format: See object «BUTTON Object (BUTTON) »

Save Data: See object «BUTTON Object (BUTTON) »

Layout Category

See object « LABEL »

Action Category

onClick: See object «BUTTON Object (BUTTON) »

OnMouseOut: See object «BUTTON Object (BUTTON) »

OnMouseOver: See object «BUTTON Object (BUTTON) »

Miscellaneous Category

See object «BUTTON Object (BUTTON) »

Tips and Tricks

5.2.7 IMAGE Object

With this object, you can display an image on the screen.

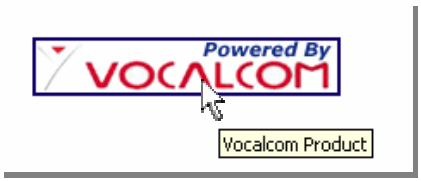
Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

AlternateText: indicate the text to be displayed instead of the image when it is not loaded, as well as the text displayed on mouse-over.



In this example, **AlternateText** is “Vocalcom Product”.

Back: See object “LABEL Object”
Color: See object “LABEL Object”
Style: See object “LABEL Object”
Width: See object “LABEL Object”
Image: See object “LABEL Object”
CSS Class: See object “LABEL Object”

Behavior Category

AccessKey: See object “LABEL Object”

Visible: See object “LABEL Object”

ToolTip: See object “LABEL Object”

Test Format: See object « BUTTON Object (BUTTON) »

SaveForm: See object « BUTTON Object (BUTTON) »

Layout Category

See object “LABEL Object”

Action Category

onClick: See object « BUTTON Object (BUTTON) »

onMouseOut: See object « BUTTON Object (BUTTON) »

onMouseOver: See object « BUTTON Object (BUTTON) »

Miscellaneous Category

See object « BUTTON Object (BUTTON) »

Tips & Tricks

5.2.8 DROP-DOWN LIST Object (DROPDOWNLIST)

With this object, you can create a drop-down list with static or dynamic values.

Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

Back: See object “LABEL Object”

CSS Class: See object “LABEL Object”

Font Category

See object “LABEL Object”

Behavior Category

AccessKey: See object “LABEL Object”

TabIndex: See object « TEXT Object (TEXTBOX) »

Visible: See object “LABEL Object”

Enabled: See object “LABEL Object”

Data Category

See object “LABEL Object”

Value Col.: Defines the column whose content must be used as object value. See “LABEL Object” to get details about the use of the Connections dialog window.

Layout Category

Left: See object “LABEL Object”

Top: See object “LABEL Object”

Width: See object “LABEL Object”

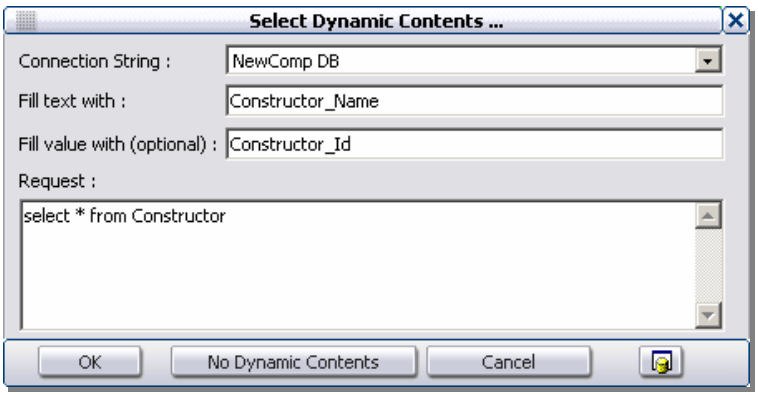
Z-Index: See object “LABEL Object”

Action Category

OnChange: See object « TEXT Object (TEXTBOX) »

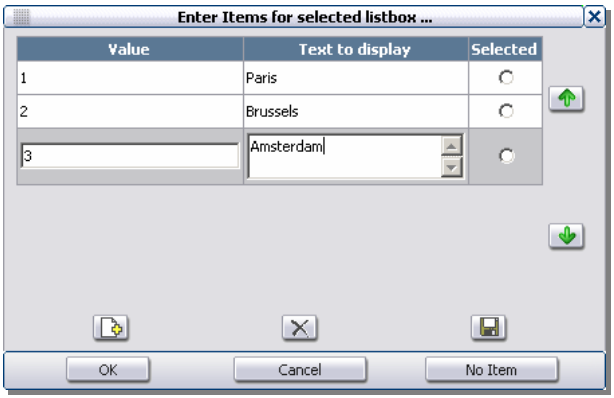
Miscellaneous Category

DynamicList: Allows you to define a request on a database and to use the result to populate a drop-down list. By clicking on the button, you get the window to define the connection.



You must choose the connection chain that defines the database in use, the field retrieved by the query to populate the list (text and value), and the SQL query used to select the records. The button can be used to open the **Connection Link** screen. See chapter « Connection Link » on page 142 for more details.

Static Items: Define static list content. By clicking on the button, you get an input window where you can list items.



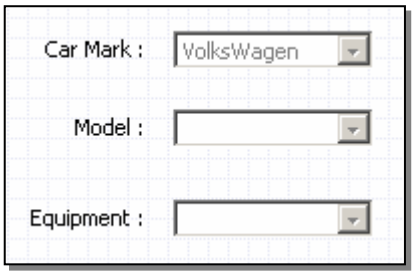
A new item can be created with the button by entering the values in the following columns: **Value** indicates the value to be saved and associated with the selected field, and **Text to display** indicating what will appear in the drop-down list on the screen.

Use the and buttons to change the position of an item. Use the button to delete the selected suggestion. The selection buttons in front of each line allow you to define which item is selected and visible by default in the list when displayed on the screen. Click on No Item to delete all other items shown in the window.

ExportOrder: See object “LABEL Object”

Tips and Tricks

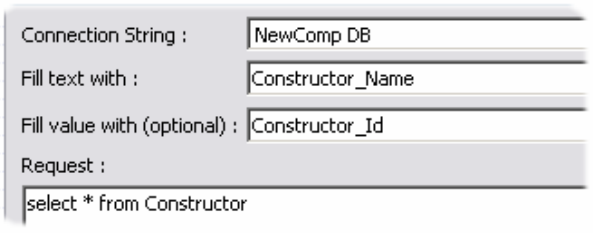
Cascade lists (content defined by the selected item in an other list).



Example :

The list of models depends on a car mark, and equipment depends on a model.

We will use the **DynamicList** property to define lists content.



First list content :

The content of the 2 other lists will be defined using a query containing a global variable. This variable is updated with the content of the previous list:

Models List query :

Select * from Model where Constructor_Id = $constructor

Series List query :

Select * from Serie where Model_Id=$model

The 2 variables $constructor and $model must be initialized at the first loading of the page.

Synchronization of the lists:

First, we create an action linked to the « onChange » event of the first list. This action is composed of 2 commands. The first is a custom function :

$constructor=@MarkList.value ;

And the second command goes back to the same script page in order to refresh its content.

Afterwards, we create an action linked to the « onChange » event of the second list. This action is composed of 2 similar commands. The first is a custom function:

$model=@ModelList.value;

And the second goes back to the same script page.

How to get the text of an item from a list.

@ListName.options[@ListName.SelectedIndex].text ;

5.2.9 LIST Object (LISTBOX)

With this object, you can create a list of values that are defined to be static or dynamic.

Description of Properities

Name Category

See object “LABEL Object”

Appearance Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Font Category

See object “LABEL Object”

Behavior Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

SelectionMode: Indicates if multiple choices are allowed.

Data Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Layout Category

See object “LABEL Object”

Action Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Miscellaneous Category

DynamicList: See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Static Items: See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Rows: Indicates the number of lines displayed in the list.

ExportOrder: See object “LABEL Object”

5.2.10 CHECKBOX Object

With this object, you can create checkboxes linked to labels. You can select multiple boxes at once, contrary to the radio buttons described in the next paragraph.

Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

Alignment: See object “LABEL Object”

Back: See object “LABEL Object”

Color: See object “LABEL Object”

Style: See object “LABEL Object”

Width: See object “LABEL Object”

CSS Class: See object “LABEL Object”

Font Category

See object “LABEL Object”

Behavior Category

AccessKey: See object “LABEL Object”

TabIndex: See object « TEXT Object (TEXTBOX) »

Visible: See object “LABEL Object”

ToolTip: See object “LABEL Object”

Enabled: See object “LABEL Object”

Data Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Layout Category

See object “LABEL Object”

Action Category

onClick: See object « BUTTON Object (BUTTON) »

Miscellaneous Category

Static Items: See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

ExportOrder: See object “LABEL Object”

5.2.11 RADIO Object (RADIOBUTTON)

With this object, you can create radio button boxes linked to labels. Contrary to checkboxes, these buttons can only be selected one by one.

Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

See object « CHECKBOX Object »

Font Category

See object “LABEL Object”

Behavior Category

See object « CHECKBOX Object »

Data Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Layout Category

See object “LABEL Object”

Action Category

See object « CHECKBOX Object »

Miscellaneous Category

See object « CHECKBOX Object »

5.2.12 OBJECTION Object

With this object, you can create a list of objections defined with static or dynamic values. This object differs from the other list-object types in its setup: an object of the drop-down list type is associated with a text zone; the content of this zone is adapted dynamically in function of the value selected in the first list.

Description of Properties

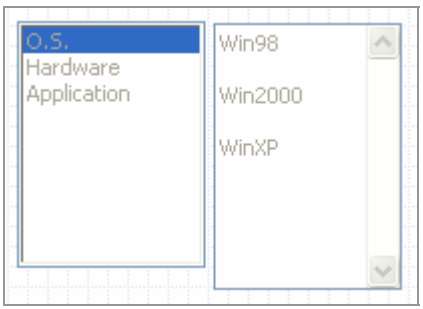
Name Category

See object “LABEL Object”

Appearance Category

Alignment: Applied to the Objection object, the alignment allows you to choose the presentation of the two fields that compose the object. A vertical alignment shows the fields on top of one another.

A horizontal alignment shows the fields next to one another.



Color: See object “LABEL Object”

Style: See object “LABEL Object”

Width: See object “LABEL Object”

CSS Class: See object “LABEL Object”

Font Category

See object “LABEL Object”

Behavior Category

AccessKey: See object “LABEL Object”

TabIndex: See object « TEXT Object (TEXTBOX) »

Visible: See object “LABEL Object”

ToolTip: See object “LABEL Object”

Data Category

See object « DROP-DOWN LIST Object (DROPDOWNLIST) »

Layout Category

See object “LABEL Object”

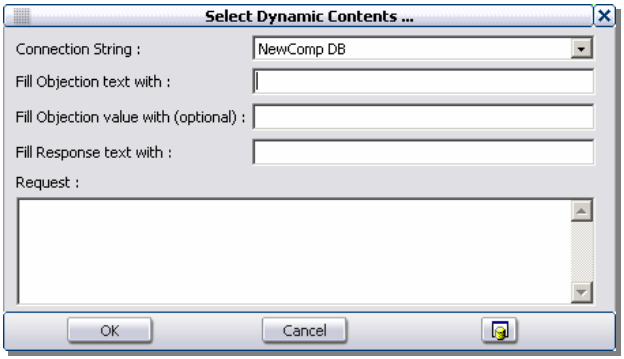
Miscellaneous Category

DynamicContents: Allows you to define a request on a database, the result of which will be used in the objection list. See chapter « Connection Link » on page 142 for more details.

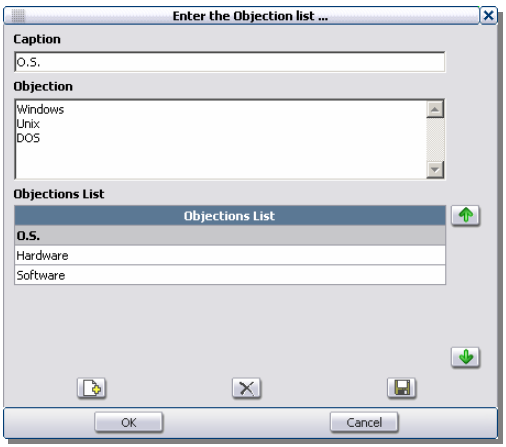
The button can be used to open the window **Select Dynamic Contents**. First choose the connection string used to connect the right database, the result fields

retrieved by the SQL statement used to populate the object content (objections text, objections value and responses value), and the SQL statement used to get data.

The button can be used to open the **Tables properties** window. (see chapter "Connection Link" on page 142)



Objection List: allows you to define the static content of the objection list. By clicking on the button, you get an input window with the different elements of the list.



A new element is created by entering the values in **Caption** and **Objection**, and then clicking on the button. Use the valid entry. Use the and buttons to change the position of an item. With the button you can delete the selected objection. Click on to cancel the last modifications and close the window.

ExportOrder: See object “LABEL Object”

5.2.13 The IFRAME Object

With this object, you can create an Internet navigation zone on the screen called IFrame.

Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

FrameBorder: Indicate whether the border of the Iframe is visible or not.

CssClass: See object “LABEL Object”

Behavior Category

TabIndex: See object « TEXT Object (TEXTBOX) »

Scrolling: Indicate if the Iframe object's scroll bars are displayed or not.

Auto: the scroll bars are displayed in function of the content.

No: the scroll bars are not displayed.

Yes: the scroll bars are always displayed.

Layout Category

See object "LABEL Object"

Margin Width: defines the width of the window's left margin.

Margin Height: defines the height of the window's top margin.

Miscellaneous Category

URL: indicates the URL of the Web page that will be displayed in the Iframe window. Click on the button to open an input dialog box.

ExportOrder: See object "LABEL Object"

5.2.14 DATAGRID Object

With this object, you can create a table, the content of which will be the result of a request on the database.

Description of Properties

Name Category

See object "LABEL Object"

Appearance Category

TextAlign: Indicates the alignment of the text in each of the table's cells (left, right, centered).

Color: See object "LABEL Object"

Style: See object "LABEL Object"

Width: See object "LABEL Object"

CSS Class: See object "LABEL Object"

Image: image to display in the background of the table.

BgColor: background color of the table cells.

Alternate: background color of the cells for one out of two lines in the table.

Selected: background color of the selected line.

Over: color of the line when the cursor moves over it.

Header: color of the table's header line.

Selected: color of the text on the selected line.

Header: color of the text on the header line of the table.

GridStyle : name of the defined style assigned to datagrid.

Font Category

FontName: See object "LABEL Object"

Text: See object "LABEL Object"

tSize: See object "LABEL Object"

Bold: See object "LABEL Object"

Italic: See object "LABEL Object"

Underline: See object "LABEL Object"

Selected: Indicate whether the text of the selected line must be displayed in bold.

Behavior Category

Visible: See object “LABEL Object”

ShowHeader: Indicates if the table’s header is visible as the rest of the table.

Pagesize: Indicate how many lines of the table are visible on each page. If the total number of lines is more than the number of lines defined here, a page chooser and navigation arrows will appear under the table.

ToolTip: See object “LABEL Object”

Enabled: See object “LABEL Object”

Allow Sort: Indicates if the displayed columns of a table can be sorted with a mouse click on the columns header.

Data Category

Connection: See object “LABEL Object”

Request: Allows you to define a request to the database, the result of which will be used in the table. See chapter « Connection Link » on page 142 for detailed information.

Layout Category

See object “LABEL Object”

Header: defines the height (in pixels) of the table’s header line.

Line: defines the height (in pixels) of the table’s other lines.

CellSpacing: defines the spacing (in pixels) between each cell.

CellPadding: defines the padding (in pixels) of the table’s header cells.

Border size: defines the table’s border size in pixels.

Action Category

onSelect: Link an action to the event ‘selection’ on one of the table’s lines.

Miscellaneous Category

ExportOrder: See object “LABEL Object”

Tips and Tricks

DataGrid populated by a dynamic query on the script first page.

This specific case needs a special setting because the next problem occurs:

Data used to populate a datagrid are retrieved and calculated at the server level, although dynamic query is build at client level, when the page is displayed. That means that the query is not yet known at the moment of the data retrieval (the variable used to build the query is not yet initialized).

The solution is to create another page set as first script page, and to set an action on the **OnLoad** event of this page. This action will be composed of 2 commands: the first one builds the query, the second one navigates to the page containing the datagrid.

References of the datagrid in JavaScript code

The variable @<DataGridName>.line contains the content of the selected datagrid line, and @< DataGridName >.row contains the number of the selected line.

5.2.15 The LITERAL Object

With this object, you can create a text zone of the « Literal » type: this is a text area with HTML content that is interpreted at execution. You can use this object to insert an ActiveX component in the screen.

Description of Properties

Name Category

See object “LABEL Object”

Appearance Category

Text: Contains the HTML code that will be interpreted and shown on the screen.

Behavior Category

Scrolling: See object « The IFRAME ».

Layout Category

See object “LABEL Object”

Miscellaneous Category

ExportOrder: See object “LABEL Object”

5.2.16 The CALENDAR Object

Allows you to insert a calendar.

Description of Properties

Name Category

See object “LABEL Object”

Behavior Category

DisplayType: Define whether the data in the calendar should be displayed by week, work day or day.

Planning: Define the planning assigned to the calendar at startup. The list proposes all the plannings defined on the system.

Commercial : Name of the default commercial assigned to the calendar at startup. It can be a text or a variable.

Subjet : Default subject assigned to appointments created into the calendar at startup. It can be a text or a variable.

Note : Default note assigned to appointments created into the calendar at startup. It can be a text or a variable.

Data Category

All the fields listed here concern the contact of the current appointment and can be filled using text or variable.

LastName

FirstName

Adr1

Adr2
Cp
City
Country
Prof. Phone
Mob. Phone
Assi. Phone
Prof. Fax
Prof. Email
Perso. Emain
Assi. Email
Company
Function
VIP

AutoCreate : If this field is “1”, the contact assigned to the appointment is automatically stored into the Agenda database. If it’s “0”, the agent must add himself the contact into the database.

If these data are not defined at the startup of the Agenda object, they can be entered by the agent via the Agenda interface within the script.

Layout Category

See object “LABEL Object”

Actions Category

Apoint. Add : Indicates action to be called when an appointment is added into the agenda.

Apoint. Upd. : Indicates action to be called when an appointment is updated into the agenda.

Apoint. Del. : Indicates action to be called when a appointment is removed from the agenda.

Miscellaneous Category

Change Planning : indicates if the agent can change the planning during execution.

Add Appointm. : indicates if the agent can add appointment into the planning during execution.

Move Appointm. : indicates if the agent can move appointment into the planning during execution.

Edit Appointm. : indicates if the agent can edit appointment within the planning during execution.

Del. Appointm. : indicates if the agent can remove appointment from the planning during execution.

ExportOrder: See object “LABEL Object”

6 COMPOSITION OF A SCRIPT

In this chapter, we discuss the main stages in the creation and configuration of a script and we provide an overview of user screens and the application's major functionalities.

6.1 How to generate a script

Before you can start creating scripts, the administrator must create an entity « Company ». Once this is done, he can create accounts for each of the company's users, who can then launch the Web Scripter application and log in.

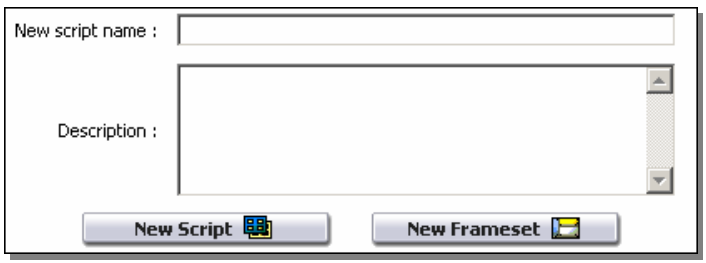
6.1.1 Create a new script

Start creating your script in the script management window.

Click on the menu bar **Create a new script** to start a script from a blank screen.

The functions **Display Script List** and **Restore a script from a file** allow you to open or restore an exported script (see sections « Scripts List » and « Restore scripts » on page 19 and 22 for more details)

Enter the name of your new script in the **New script name** field, and a description:



Click on

to generate the script and launch the editor.

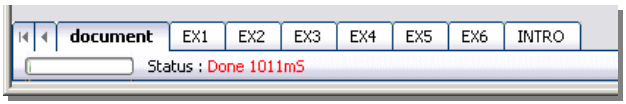
6.1.2 How to manage script pages

An empty page named **Index** appears. It is created by default and defined as the first page of the script. You can rename it, duplicate it or create other empty pages.

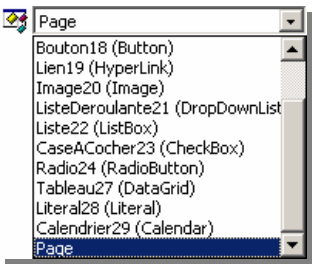
You can also import pages that have been stored as templates in the **Template** menu, thus adding predefined pages to your script.

See section « Template Menu » on page 29 for more information on the options.

To navigate between pages of your script, click on navigation tabs displayed at the bottom of the screen.



Or via the objects list just under the toolbar.



In the next step, you will change the page layout and add different objects.

6.1.3 Page layout

You can change the page layout with attributes such as title, size, color, and more. You can even put an image in the background of your page. (See chapter « PAGE Object » on page 67 for more information on changing the look of your script page.)

Thanks to the **onLoad** property, you can initiate actions when it is started up. (See chapter «Actions and Commands » and « Description of the possible commands » on page 152 and 157 for more information on creating and managing actions)

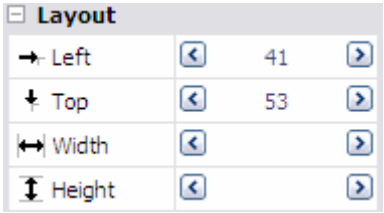
6.1.4 Create objects on the script page

In this phase, you will add objects to the displayed page and change their attributes to get the desired result.

To add an object, use the toolbar in the object zone.

The list of objects is available in chapter « The Objects » on page 67.
By clicking on the button of the object you wish to add, it is loaded onto your page and given a default name and number in line with its type and creation order. For example, the first object created is of the **Label** type. It will be named « Label0 ».

Put it in the desired place by holding the mouse on the button and by changing the defined values for the attributes **Left** and **Top** in the **Layout** sub-menu.



Change the size with the small arrows around the object or by changing the values of the **Width** and **Height** attributes.
(For more details, check out the section « Select an Object » on page 44).

You can also change the order of the objects through the main menu (By default, the object last created comes on top of the ones previously created) and align them.
To perform these changes on several objects at once, you can select them all by framing them with the mouse or by clicking on each individual object while holding down the Ctrl key.

Once you have selected them, you can create a group of objects with the **Group objects** function in the **Edition** menu. This simplifies all further changes you want to execute for a group of objects

The properties in the left menu are adapted in function of the type of object selected. In case of multiple selection, changing a common attribute causes to change it in all the objects that contain this particular quality.

In order to change the context of the text displayed in or on the object, you have to change the value of the **Text** attribute in the sub-menu **Appearance**.

You can also change an object's look by changing its properties with regard to color, border, text, shape and content.

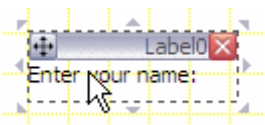
6.1.5 Text format via cascading style sheet

You can use a cascading style sheet to simplify giving the object the desired look. It allows you to define how the objects will look on the screen thanks to a style sheet external to the script. You can easily change the look of your objects by changing the external style sheet rather than having to change each page and object separately.

A cascading style sheet can be integrated in the script with the load function from the **Upload Center**. Once integrated, you merely need to link the name of the style sheet with the object in question through the **CssClass** attribute.
(See chapter « Upload Center » on page 135).

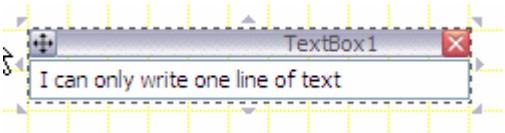
6.1.6 Text zones and lists

The object types that allow you to display text on the screen contain the following properties and must be chosen in function of what you wish to see on the screen:



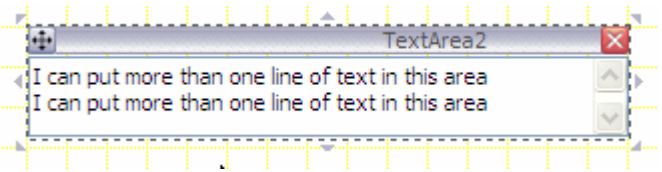
Labels are used to display permanent text on the screen. A label can be linked to another object with the property **Text for**: simply click on the label to get to the linked

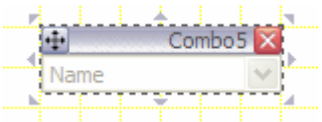
object. Also we can specify the name of a global variable as content, its content will be displayed at runtime.



Text objects allow you to display and enter text on one line. You cannot insert more lines.

Text zones allow you to display multiple lines of text. By resizing the zone or by entering text, the text will adapt itself so that it always appears completely. The scroll bar on the right side will be enabled so that you can see the whole text.

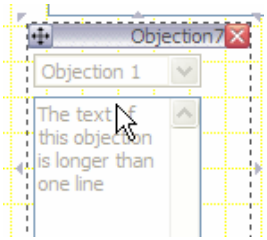




The **Dropdown list** shows one visible element on a single line. By cliking on the button, you see the other possible items to choose from. The value you choose is displayed on the single line that is shown when you close the list. You can't enter text in the scroll list as the assignment of values is done in the data entry window that you see when clicking on in front of the properties **Dynamic List** or **Static Items.** Dynamic data come from a database table; static data are entered directly in the entry field. When you have defined the content, you will see « [BINDED] » appearing in front of the property in question. Before having defined the content, this field indicated « [NONE] ».

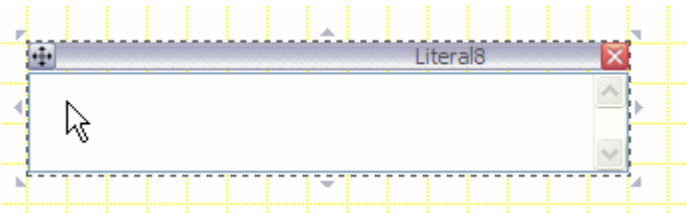
(See section «DROP-DOWN LIST Object (DROPDOWNLIST) » on page 90 for more details).

A **Listbox** allows you to display several items; the selection is done by clicking on the desired item. You can select multiple lines simultaneously, contrary to the **Dropdown List**. Its content is similarly defined with dynamic or static data, following the same principle as the drop-down list. With its **Lines** property you can define the height in terms of numbers of lines (not in pixels).



Finally, the **Objection** object shows a number of elements organized hierarchically: if you choose a superior element of the drop-down list, the field content of the inferior zone will be updated. Again, the data is created in a dynamic or static fashion.

An object of the **Literal** type is a text zone in which you can copy HTML code. It will be executed upon display of the page.

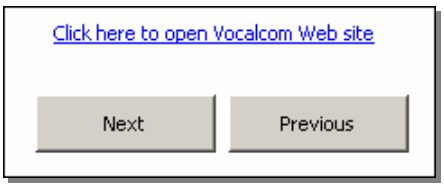


The content of all these fields can be linked to database by defining the link and executing a SQL request.

*The value of an object is referenced with the syntax @**<name object>.value**. You can insert a reference in a field by checking the **Evaluate** box in the properties of the field to see how the content will be interpreted when executed.*

6.1.7 Links and buttons

These 2 objects can be added to the screen to allow browsing to another page or to an external URL. (See chapter « Link Builder » on page 136 for information on defining links).



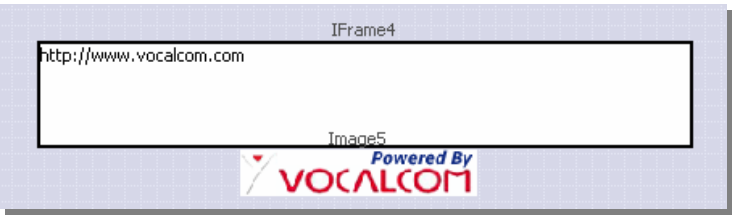
They also present on the screen a so-called action launcher object. (See chapter « Actions and Commands » on page 152 for more information on actions and commands).

6.1.8 Images and IFrames

An Image object allows you to insert an image from a private or global image list.To be displayed in the list, an image must be loaded onto the server in the **Upload Center**.
An image can also function as a link thanks to its properties **Link**, **External URL** and **Target**. (See chapter « Link Builder » on page 136 for information on defining links).

You can also assign an action or command to an image (See chapter « Actions and Commands » on page 152 for information on possible actions and commands).

An Iframe object allows you to integrate an external URL in the active page.
Example : The picture hereunder shows objects during script edition : a Iframe box containing the Vocalcom Web site URL, and an image.



The next window shows the result :

*The URL contained in the Iframe can be determined dynamically in function of another object on the page: simply assign the desired URL to the variable @**<name Iframe>.src**.*

Example: @MyIFrame.src=@DropDownList.value

6.1.9 Checkboxes and radio buttons

With these objects, the user can create single and multiple choices on the screen in the form of checkboxes (multiple choices) and radio buttons (single choice).
Their content is purely static and entered with the data entry window available when clicking on the button. (See sections « CHECKBOX Object » and « RADIO Object (RADIOBUTTON) » on page 96 and 98 for more details).

6.1.10 Datagrid

With the Datagrid object you can display structured data resulting from SQL requests. Both its layout and colors can be customized and you can link an action to the selection of a cell in the table.

*In the **Request** property, you can indicate the name of a global variable that contains a dynamically created SQL request.*

Example : Datagrid filled by a query

Result :

indice	nom	adresse
1	ABADIE JEANNEttttt	1 RUE DES EBISOIRES
2	ABAYACHOU MOHAMED	10 RUE DU PORTUGAL
3	ABBOU MOHAMMED	55 RUE DES EBISOIRES
4	ABDELKADER DJILALI	15 RUE DU PETIT BONTEMPS
5	ABDELKADER MOUSSA	15 RUE DU PETIT BONTEMPS
6	ABDEREMANE YOUSSOUF	536 RUE ABEL GUYET
7	ABDERRASAK MOHAMED	1 ALLEE SAINT JUST
8	ABDERRAZAK LATIFA	7 ALLEE SAINT JUST
9	ABDESLEM EL MOSSLIH	8 ALLEE SAINT JUST

6.1.11 The calendar

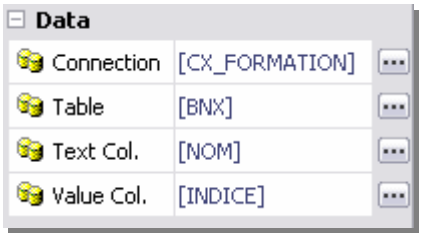
With the calendar object, you can display a dynamic agenda on the screen for managing commercial appointments and contacts.
The agenda initialization parameters (planning, commercial) can be defined during the script edition, but can also be chosen by the agent during script execution.

Access rights to the Agenda object are defined during the script edition. It is so possible to avoid an agent changing the agenda planning, or to add, modify, move or remove appointments within the agenda.

It is also possible to link actions to events generated by this object: events are appointment creation, update or suppression.

6.1.12 Create connections with databases

Certain objects can be populated by specific data coming from databases. This means that, when the script is executed, a link to the database is created and gets in real time values from the table to populate script objects. A link between an object and a database is defined with the following properties.



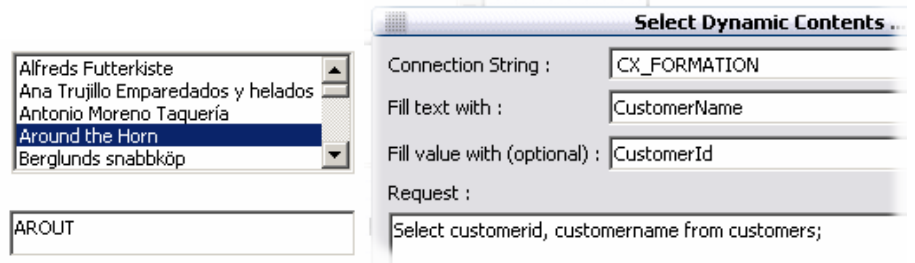
- Connection: indicate the connection previously defined in the « Connection Link » window. (See chapter « Connection Link » on page 142)
- Table: indicate the name of the table that is being used to extract data.
- Text Col.: indicate the column from which the value is extracted.
- Value Col.: For objections, drop-down and other lists, you can indicate the column from which the value is extracted.

The data used during the script execution are always related to the current index value.

The predefined variable $INDICE can, at any moment, give you the number of the current record index.

*An object's value is referenced with the syntax @**<nom objet>.value**. In case of a list, drop-down list or objection, the value property contains the choice and not the content of the list.*

Lists, Dropdown Lists, Objections and Tables have a specific property that allows them to indicate an SQL request to be performed in real time during script execution
In this example, the superior list is complemented with the following parameters:



- Connection String: indicate the previously defined connection in the « Connection Link » window. (See chapter « Connection Link » on page 142)
- Fill text with: indicate the name of a field selected by the request that serves to populate the list.
- Fill value with: indicate the name of a field used to provide the value of the list
- Request: to edit the SQL request.

The text field underneath is dynamically completed with the following parameters:

An action called « update » is assigned to the event **onChange** of the list, and the action « update » is defined by the following personalized function:

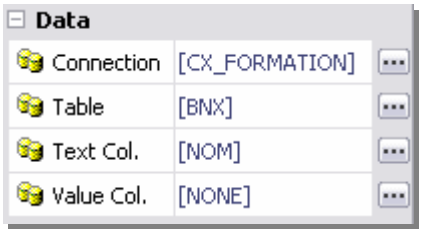
@MyTextBox.value=@MyListbox.value

Each time a user clicks on the company name in the list, the ID of the client appears in the text field.

6.1.13 Create the Client file

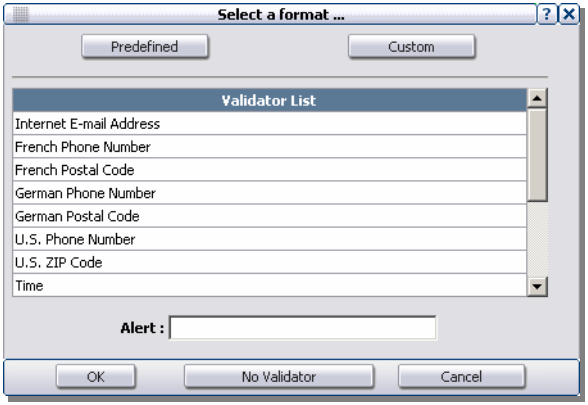
Your script shows objects whose content can be used to populate a client file.

Make sure that the necessary fields are on your page and link them to your Client file using the following properties:



It is essential that you respect the type of data to avoid conversion problems during script execution. A number of necessary validations and checks can be done when the data are entered in the database on the level of the page. You can also do a final validation and a validation when information is entered in each field, with the following properties:

- Allow Null: check if a field has been left empty or not before going on to the next field
- Format: associate the field with a predefined format from a list or by creating one if necessary.

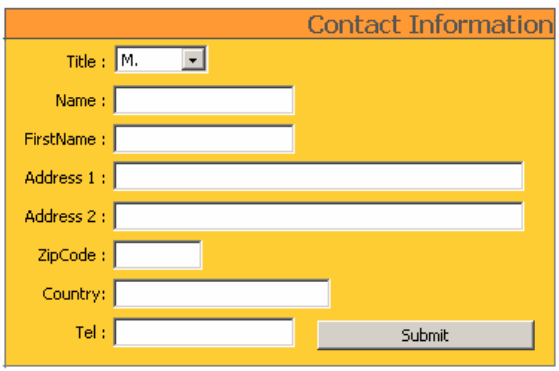


(see section "**Custom Formats help screen**" on page 52)

- Alert: indicate the error message that appears when you click on the flickering ! symbol next to the field that causes the error.

The data are added or updated physically in the destination table when you move from the active page to another one.

The data are also recorded when you click on enter on the active page.

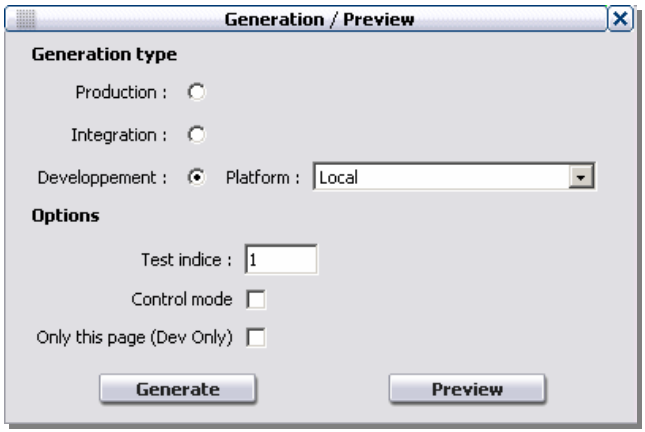


In this example, the « Submit » button is only used to reach the current page of the script. Pushing this button validates and records the data.

6.1.14 Save and execute a script

When your page is finished, you have to save the script on the server. You can also configure the application to perform an auto save (every chosen delay).

In the **Pages** menu, indicate which page is the first page of the script. Once you've finished and linked all pages, the entire script is ready for test and execution:
Use the **Generation** function in the **Script** menu to verify if the generated code is correct. The next window appears :



First you have to choose the generation type : Production, Integration or Developpement. In the case of the script should already be used by real agents in a production environment, it's recommended to first generate and test the script in the Dev. environment **(Dev. version)**.

We can also choose the **Platform** to run the script.
A test step can also be useful : use the Integration type.
When it's correct, it can be generated in the production environment **(Run version)** in order to replace the script used by the agents.

In the **Options** part, you can select the **Test Indice**. (default is 1)

☑ **Control mode** : It is reserved for advanced users : Checking this box allows showing the debug window during the execution of the script, or during the preview. (see section "Debug Window" on page 129 to read details about debug possibilities).

☑ **Only this page (Dev only)** : allows testing only the current page, and not the whole script. This is only possible in development mode.

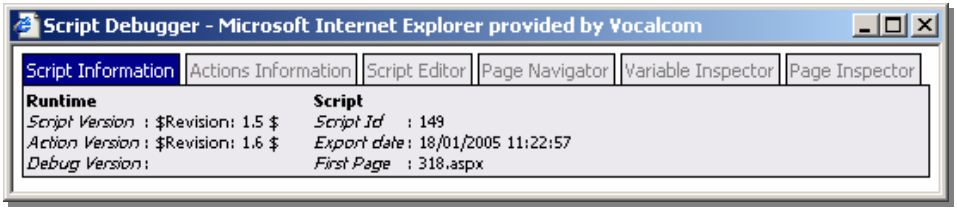
Generate This button starts compilation and generation of the script. Once it's ready, it's executed.

Preview this button executes the result of the last script generation.

6.1.15 Debug Window

During script execution, it's possible to consult the debug window to check variables content. This is possible only if you checked the **Control Mode** box on the **Generation/Preview** window.

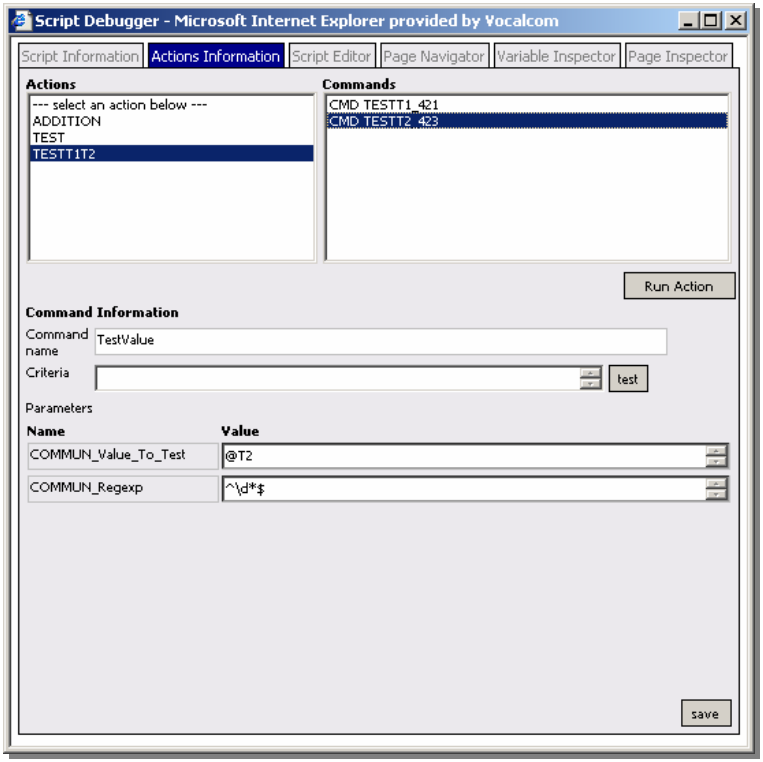
In the upper-left corner of the script window, a symbol is shown.
If you click on it, a separate debug window appears.



It allows accessing the next functionalities via the tabs :

- **Script Information** : This window displays general information line script version and revision number.

- **Actions Information** : This window is very useful because you can access all the commands and actions defined into the current script, to consult actions parameters and to execute it separately.



The upper-left list displays all the actions defined in the script. The right list displays commands that compose the selected action.

When an action and a command are selected, the parameters of the command are displayed in the next part named **Command Information**.

The names of the used variables are displayed in the part named **Parameters**.

The button

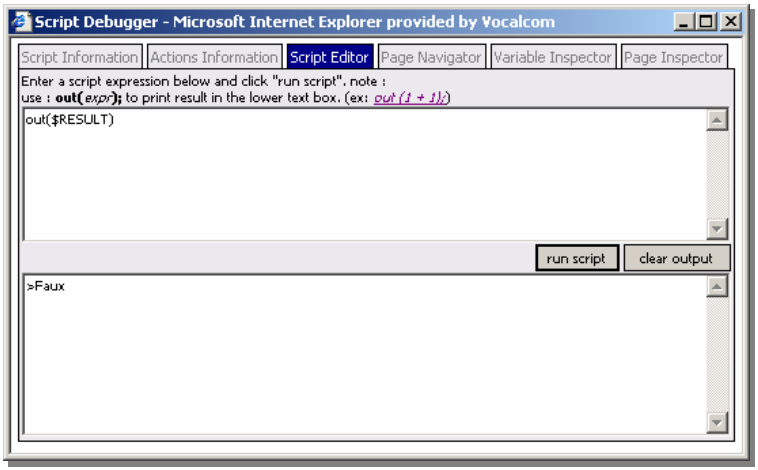
can be used to test a specific command at every time.

The button

can be used to test execution conditions.

- **Script Editor** : The upper part of the window can be used to type JavaScript code.

Example : the command **out(**<variable>**)** can be used to display the content of a global variable or a script object.


Script Debugger - Microsoft Internet Explorer provided by Vocalcom
Script Information
Actions Information
Script Editor
Page Navigator
Variable Inspector
Page Inspector
Enter a script expression below and click "run script". note :
use : out(expr); to print result in the lower text box. (ex: out (1 + 1);)
out($RESULT)
run script
clear output
>Faux

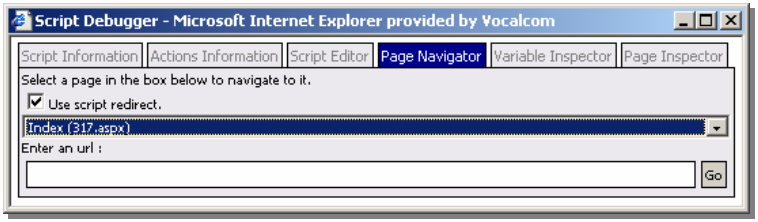
Just click on the button run script to execute the entered code and to see the result in the lower zone. The button clear output can be used to clear the lower zone.

It's also possible to assign a value to a variable via this window, using the next syntax : $Variable="Hello World!";

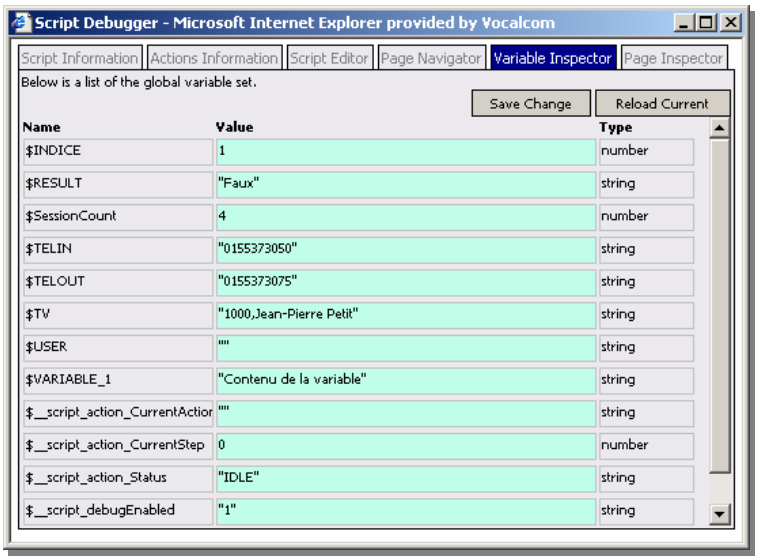
And then click on the run script button.

- **Page Navigator** : This window allows reaching directly a page of the script. In the list you will see the name of the generated pages.

It's also possible to choose an URL using the lower field and the button Go.



- **Variable Inspector** : This window can be used to consult predefined script variables. The variable type is displayed.

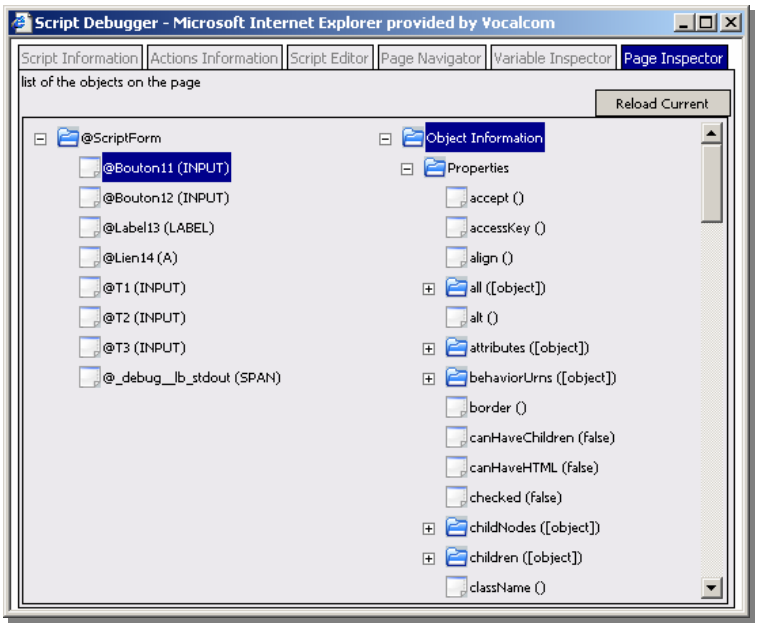


On this screen, it's possible to modify the content of variables and to apply modifications, so they are directly taken into account in the running scrip.

A modified variable is displayed in pink, you can use the Save Change button to apply the new value.

Use the button Reload Current to load and refresh the current content of the script variables.

- **Page Inspector** : This window shows graphical overview of the objects existing on the current script page. It can be used to view directly properties linked to an object.



On the left side, under **@ScriptForm**, all the objects are displayed.
On the right side, under **Object Information**, all the properties of the selected object are displayed (properties, style parameters, events).

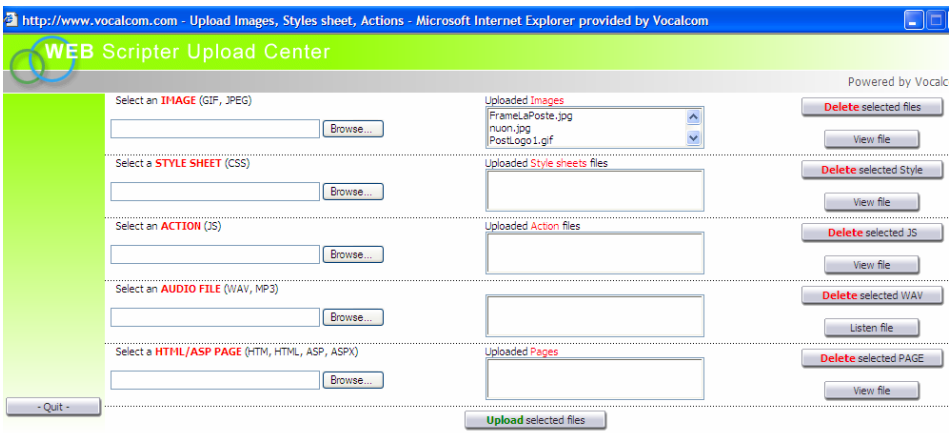
When you click on the ⊞ symbol, you open the tree with the complete list of the current object components.

The button

can be used to load and refresh the page.

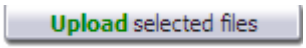
6.2 Upload Center

The **Upload Center** screen allows you to manage and use elements shared with other users from the same company. These can be images, style sheets, actions, audio files, HTML and ASP pages.

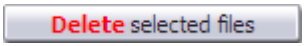


The

button lets you launch an explorer window and choose elements from your workstation that can be put at the disposal of others on the server. Once selected locally, the element's name appears in the field of the left column.

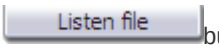
Then click on the 

button to transfer the selected elements on the server. When they appear in the **Uploaded** area, they are available to the other users.

With the 

button you can delete files from the server.

With the

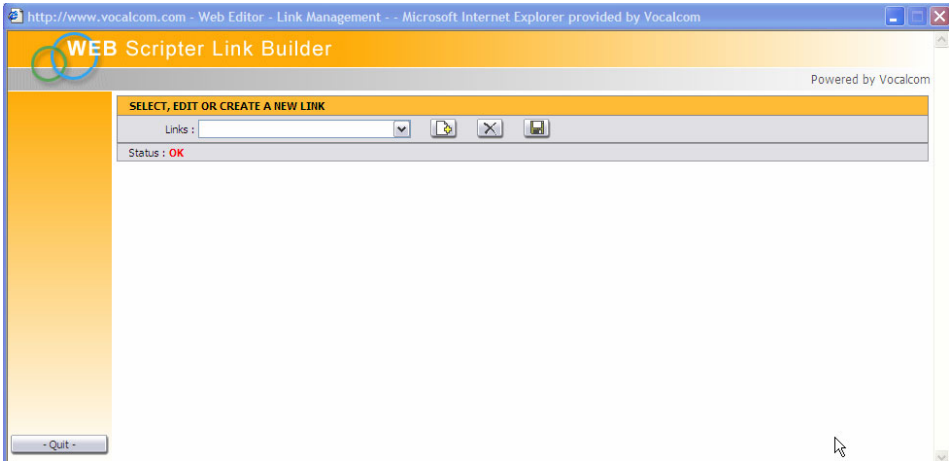
or 

button you can preview (or hear) the item selected from the list.

All elements in the **Uploaded** list are accessible during the edition phase of the script through the following functions: **Load** from the **Template** menu, the properties **Image** and **CSS Class**, or when editing specific commands and actions.

6.3 Link Builder

The window in which you can define links allows you to create predefined links that are used in the script with the **Link** property of buttons, links or images. You can determine the link's destination in function of certain conditions.

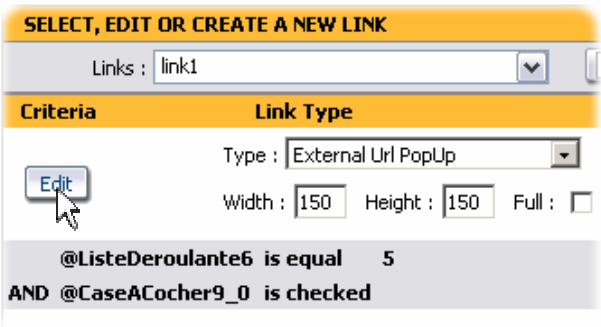

http://www.vocalcom.com - Web Editor - Link Management - - Microsoft Internet Explorer provided by Vocalcom
WEB Scripter Link Builder
Powered by Vocalcom
SELECT, EDIT OR CREATE A NEW LINK
Links :
Status : OK
- Quit -

Encode a link name in the **Links** zone and click on the button. A default criterion appears in the inferior list. Choose a primary destination; select a link type in the **Type** list:

- Previous: go to the previously displayed page
- Page script: refer to an active script page
- External URL: refer to an Internet site
- Page Script PopUp: indicate a page script that will be displayed in a dialog box of a specific size on top of the active page.
- External URL PopUp: refer to a URL whose content will be displayed in a dialog box of a specific size on top of the active page.

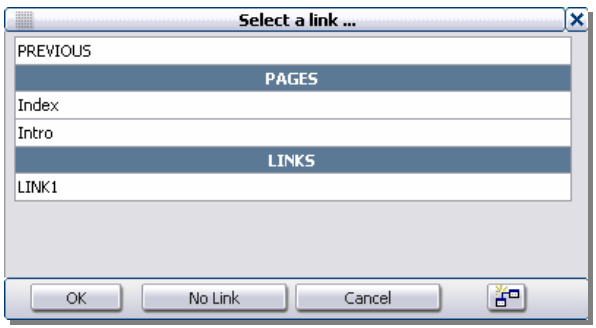
Encode the address in case of an external URL.

Then use the Edit button to launch the criteria editor. (See chapter « Criteria Builder » on page 140.). These criteria allow you to define the conditions to be evaluated in order to determine the link's destination. Once a criterion has been defined, moving the cursor over the Edit button results in the display of the condition:



Use the button to delete a destination and its condition. Use the button to add another destination and condition.

Once defined, the newly created link is shown in the list displayed by the application when you define the **Link** property.



In this window, the button allows you to open the **Link Builder** window.

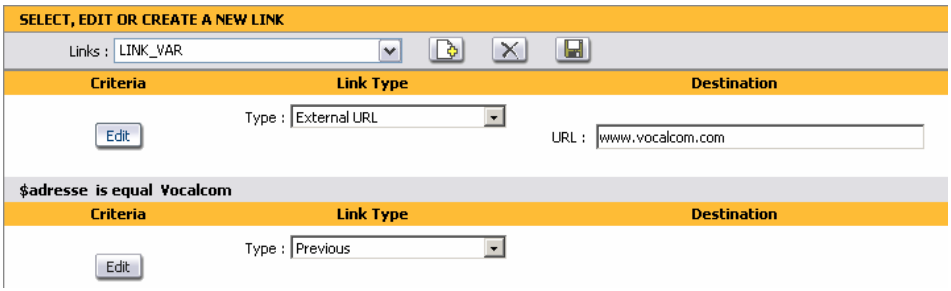
At the bottom of the **Link Builder** screen, a status line lets you verify the validity of the links.

Tip and Tricks

How to do a conditionnal branch via the Link Builder:

It's recommended to use links definition instead of actions to do a conditional branch :

On the Link Builder, create a link composed of several destinations, and for each destination, define a condition. When the condition result is true, then the script branches to the involved destination.

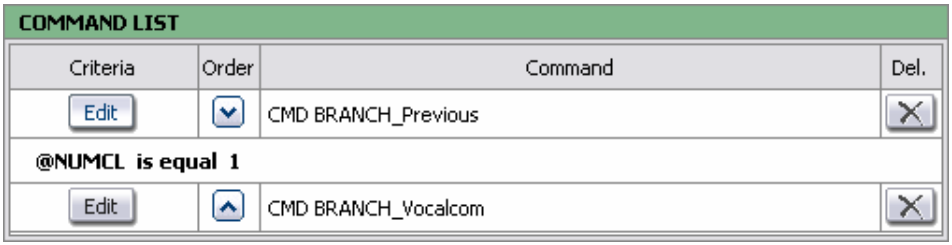


In this example, the link called « LINK_VAR » is composed of 2 destinations.

The external URL www.vocalcom.com will be used as destination if the variable $adresse equals to the value « Vocalcom ».

The script comes back on the previous page if the variable $adresse in not equal to "Vocalcom".

On the other side, a conditional branch using actions can meet the next problem : a condition with result false cancels the execution of one command, and not the other ones. Example :

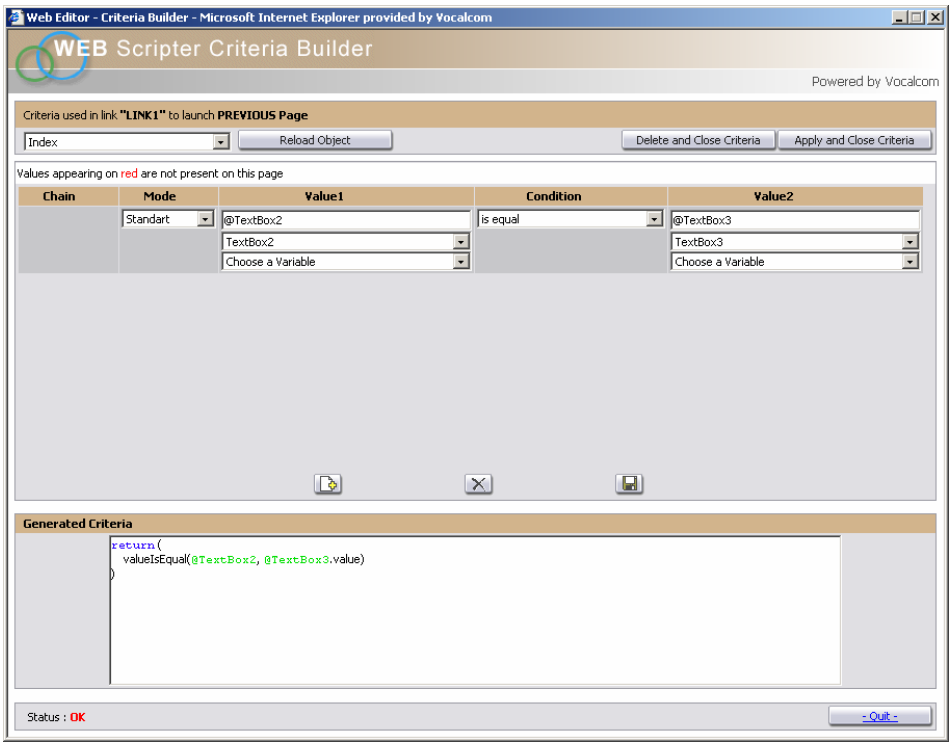


In this example, the first condition gives a true result, then the script goes to the « Previous » page.

Once on this page, the next command is executed, but she is also linked to a condition using the NUMCL object value. This object doesn't exist on the current page, and a script error happens.

6.4 Criteria Builder

In this window, you can determine the conditions to be used in the **Link Builder** and **Action Builder** screens.



The description of the link in question is displayed on the top line. With the Delete and Close Criteria button you can cancel the latest unregistered modifications and go back to the previous screen.

The Apply and Close Criteria button allows you to register the latest modifications and go back to the previous screen.

According to the selected page in the first list, the content of the suggested lists has changed: the first list **Chain** can be used to link several conditions. The second list **Mode**

can be used to switch from **Standard** to **Expert** mode. The standard mode is described hereunder, and the Expert mode shows a single edition zone where the developer can type its code.

The third list shows the current page objects : **Choose an Object**. The list right underneath shows the defined variables defined in the script. With these 2 lists you can put together the first part of the expression: **Value1**.

The central **Condition** list asks you to select an agent relative to the type and state of the selected object. The next zone **Value2** allows you to select the rest of the expression: a directly entered value, an object or a global variable.

Thanks to the button, the information is registered and displayed in the lower zone of the screen **Generated Criteria**. Click once on a criteria line to edit fields.

When the information is recorded, the button can be used to add a new condition linked to operator OR or AND. With the button you can delete a criterion.

The

button allows you to update the content of the object lists and variables by reloading them from the latest stored version of the page.

6.5 Connection Link

With the Connection Link, you can immediately view all objects of the script and their links to databases.

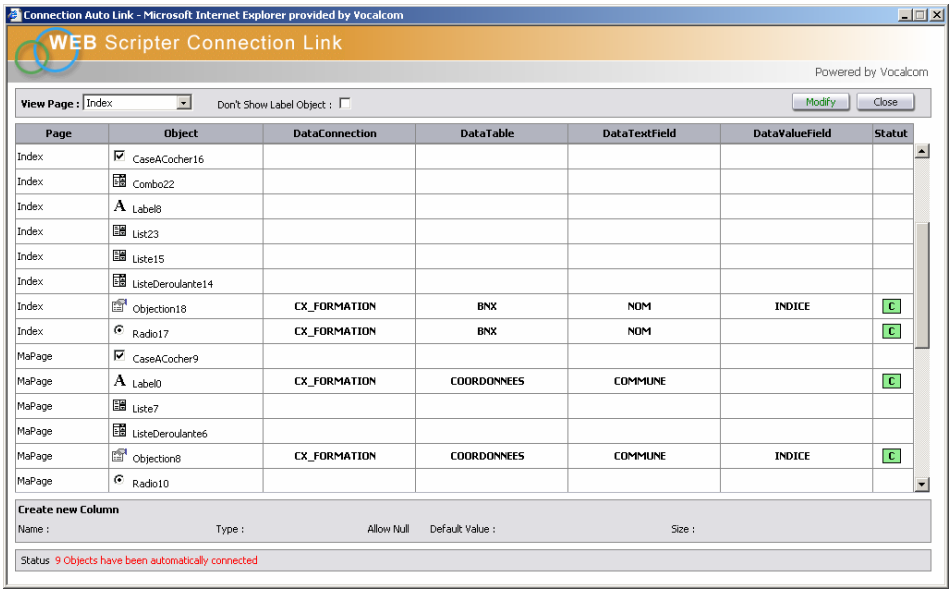
It consists of 2 main parts:

The first part displays the properties Data Connection, Data Table, Data Text field and Data Value field, as they were configured for each object; it's the « **Connection Auto Link** » window. (see section « Connection Auto Link : Start Customer File Connection » on page 144)

The second part allows displaying and generating new table columns based on script objects. It's the « **Connection Create Link** » window. (see section « Connection Create Link : Start Customer File Creation » on page 146).

6.5.1 Connection Auto Link : Start Customer File Connection

This part of the application can be used to link script fields to existing table columns.


Connection Auto Link - Microsoft Internet Explorer provided by Vocalcom
WEB Scripter Connection Link
Powered by Vocalcom
View Page : Index
Don't Show Label Object :
Modify
Close
Page | Object | DataConnection | DataTable | DataTextField | DataValueField | Statut
Index | CaseACocher16
Index | Combo22
Index | Label8
Index | List23
Index | Liste15
Index | ListeDeroulante14
Index | Objection18 | CX_FORMATION | BNX | NOM | INDICE | C
Index | Radio17 | CX_FORMATION | BNX | NOM | | C
MaPage | CaseACocher9
MaPage | Label0 | CX_FORMATION | COORDONNEES | COMMUNE | | C
MaPage | Liste7
MaPage | ListeDeroulante6
MaPage | Objection8 | CX_FORMATION | COORDONNEES | COMMUNE | INDICE | C
MaPage | Radio10
Create new Column
Name :
Type :
Allow Null
Default Value :
Size :
Status 9 Objects have been automatically connected

The **View Page** list allows direct displaying of objects on a specific page. The checkbox **Don't show Label Objects** allows filtering objects and not displaying labels.

First columns on the links display the object name and page; next columns **Dataconnection, DataTable, DataTextTable, DataValueField** allow you to consult the values assigned in the same-name properties of the objects.

In the **Statut** column, you can check the object connection state.

C : the object is already connected to a DB column.

A : the application found an existing column in the DB with the same name than this object and suggests connecting it.

The number of objects found is displayed in the status bar at the bottom of the window.

: the use just add this connection, which will be effective after he validates it.

The

button at the top rights of the window enables you to validate the new connections.

It is thus possible to use this screen to modify, add or remove links between script objects and DB columns. It is not necessary to open and modify each script page and object.

Connect an object to a DB column

First select an object line in the list.

When a line is highlighted, new lists appear in the columns and allow selecting next fields :

DataConnection : shows the list of defined connections. Select the connection to link the right DB.

DataTable : shows the list of all the selected DB tables. Select the table where the field to link is.

DataTextField : shows the list of all the columns detected in the selected table. Select the column to be linked to the field. The button can be used to create a new column into the current table. The parameters of this column are displayed at the bottom of the window :

Once all the parameters are set, click on the button

to validate and register modification. The modification will be physically applied to the database when the button

will be used.

The same way, **DataValueField** can be used to link a DB field to the value field of a list object. The same button Can be used to create a new column.

The symbol

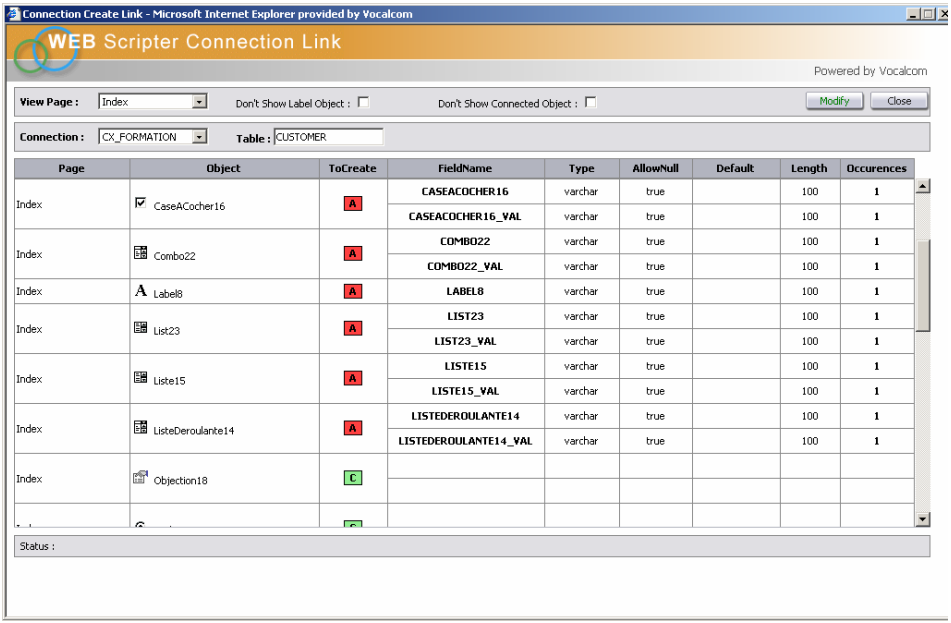
appears in the **Status** column until the button

is used.

6.5.2 Connection Create Link : Start Customer File Creation

With this application, you can create a database table based on the objects found within the script pages.

The **View Page** list allows direct display of objects on a specific page. With the checkbox **Don't show Label Objects** you can filter objects and decide not to display labels.


Connection Create Link - Microsoft Internet Explorer provided by Vocalcom
WEB Scripter Connection Link
Powered by Vocalcom
View Page : Index
Don't Show Label Object :
Don't Show Connected Object :
Modify
Close
Connection : CX_FORMATION
Table : CUSTOMER
Page | Object | ToCreate | FieldName | Type | AllowNull | Default | Length | Occurences
Index | CaseACocher16 | A | CASEACOCHER16 | varchar | true | | 100 | 1
CASEACOCHER16_VAL | varchar | true | | 100 | 1
Index | Combo22 | A | COMBO22 | varchar | true | | 100 | 1
COMBO22_VAL | varchar | true | | 100 | 1
Index | Label8 | A | LABEL8 | varchar | true | | 100 | 1
Index | List23 | A | LIST23 | varchar | true | | 100 | 1
LIST23_VAL | varchar | true | | 100 | 1
Index | Liste15 | A | LISTE15 | varchar | true | | 100 | 1
LISTE15_VAL | varchar | true | | 100 | 1
Index | ListeDeroulante14 | A | LISTEDEROULANTE14 | varchar | true | | 100 | 1
LISTEDEROULANTE14_VAL | varchar | true | | 100 | 1
Index | Objection18 | C
Status :

First columns on the links display the object name and page.

Next columns **FieldName, Type, AllowNull, Default, Length** allow you to consult the properties of the DB field related to the object.

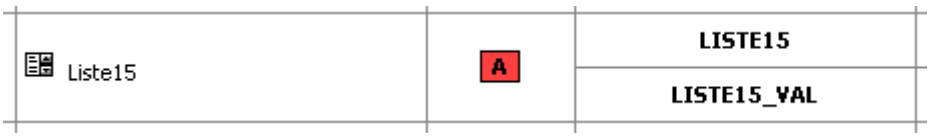
With the **ToCreate** column you can check if the connection already exists.

C : the column already exists and is connected to this object.

A : the application suggests to create the column with the next validation and to link this object. Properties values are suggested by the application by default, together with a field name based on the existing object name.

By selecting a line on the screen, the **ToCreate** column shows a checkbox where you can indicate if you want to create this DB field or not.

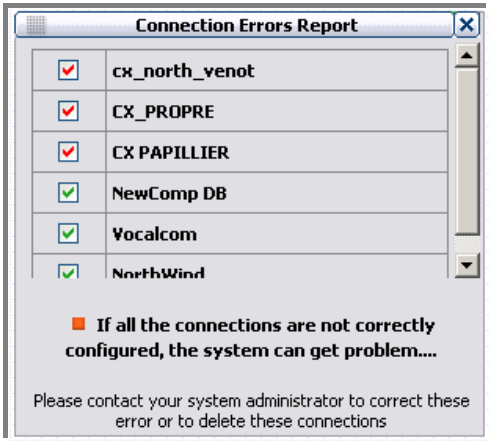
Objects already linked to a DB field also display this checkbox, meaning that you can add a supplementary DB field linked to this object. The application suggests default values for the new field to be created.
For Listbox, Dropdown List or Objection objects, the line is composed of 2 rows, the first of which contains the text used to fill the displayed list, and the second contains the value associated to this object.



The **Occurrences** column indicates the number of identical fields to be created in the DB based on this object.

6.5.3 Invalid connections check at script opening

When the user opens a script, connections known on the Web Scripter server are checked, and when a connection can not be established, a pop-up called « Connection Errors Report » appears.

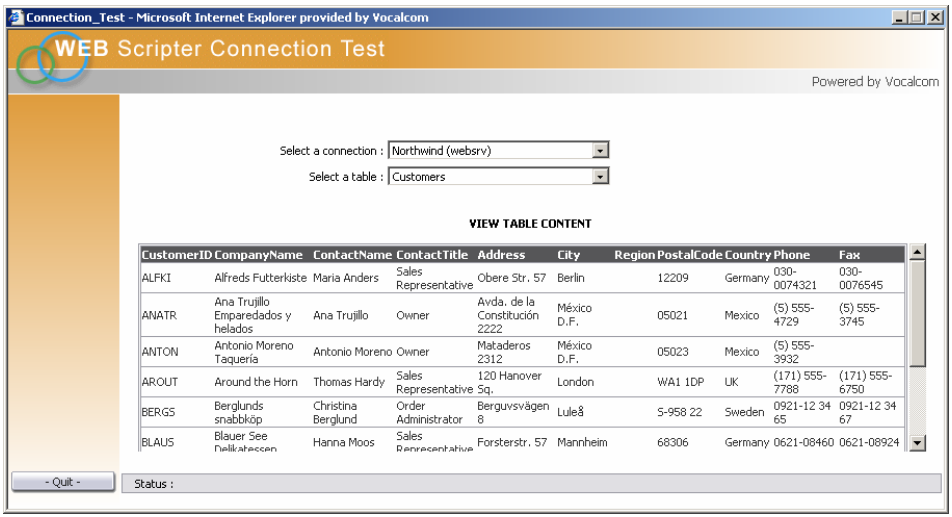

Connection Errors Report
cx_north_venot
CX_PROPRE
CX PAPILLIER
NewComp DB
Vocalcom
NorthWind
If all the connections are not correctly configured, the system can get problem....
Please contact your system administrator to correct these error or to delete these connections

This pop-up allows to see all the known connections, and to identify directly the invalid ones.

It's recommended to ask to the administrator to correct parameters, or to delete erroneous connections.

6.6 Connection Viewer

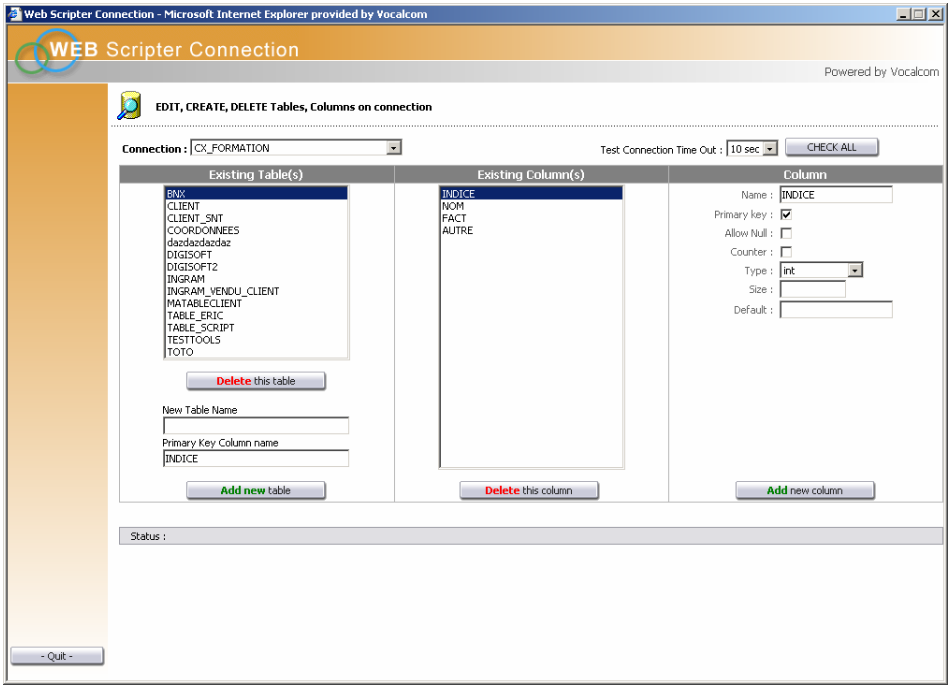
With this part of the application, you can quickly display the content of a DB table.
Start with selecting a connection in the **Select a connection** list; then select a table in the **Select a table** list.


Connection_Test - Microsoft Internet Explorer provided by Vocalcom
WEB Scripter Connection Test
Powered by Vocalcom
Select a connection : Northwind (websrv)
Select a table : Customers
VIEW TABLE CONTENT
CustomerID CompanyName ContactName ContactTitle Address City Region PostalCode Country Phone Fax
ALFKI Alfreds Futterkiste Maria Anders Sales Representative Obere Str. 57 Berlin 12209 Germany 030-0074321 030-0076545
ANATR Ana Trujillo Emparedados y helados Ana Trujillo Owner Avda. de la Constitución 2222 México D.F. 05021 Mexico (5) 555-4729 (5) 555-3745
ANTON Antonio Moreno Taquería Antonio Moreno Owner Mataderos 2312 México D.F. 05023 Mexico (5) 555-3932
AROUT Around the Horn Thomas Hardy Sales Representative 120 Hanover Sq. London WA1 1DP UK (171) 555-7788 (171) 555-6750
BERGS Berglunds snabbköp Christina Berglund Order Administrator Berguvsvägen 8 Luleå S-958 22 Sweden 0921-12 34 65 0921-12 34 67
BLAUS Blauer See Delikatessen Hanna Moos Sales Representative Forsterstr. 57 Mannheim 68306 Germany 0621-08460 0621-08924
- Quit -
Status :

The content is displayed in a scrolling datagrid.

6.7 Connection Builder and Checking

With this part of the application, you can manage table structures within the database.



First, you choose the connection in the **Connection** list at the top of the screen. This connection defines which DB to connect, with login ID and password.

The **Test Connection Time Out** allows defining the maximal waiting time before considering the connection as unsuccessful. With the

button, you can check if the DB is available at any moment.

The left-hand list shows existing tables within the DB.

The

button allows deletion of the highlighted table.

With the input fields **New Table name, Primary Key Column Name** and the

button you can create a new table and define its primary key column.

The **Existing Columns** zone shows the existing columns found in the selected table. The

button can be used to remove the highlighted column from the table.

The right part of the screen lets you create new columns in the highlighted table. Just specify in properties (type, size, default value,…) and click on the

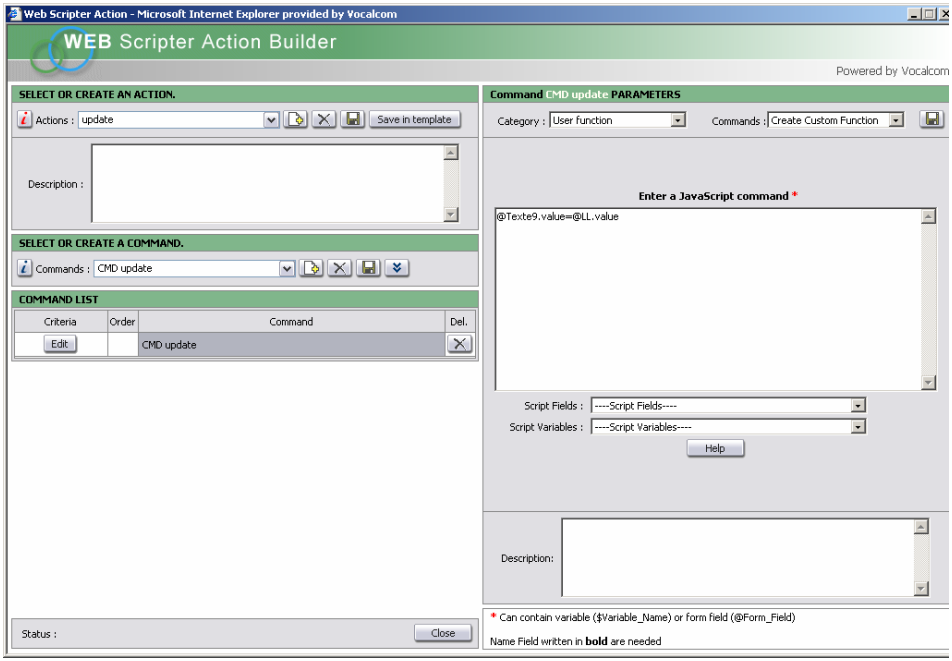
button.

6.8 Actions and Commands

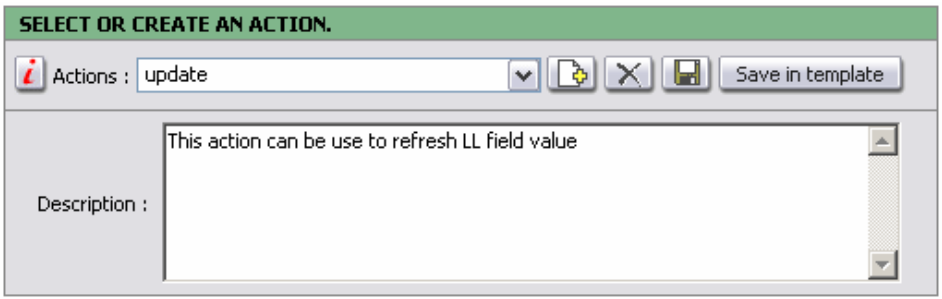
In this part of the application, you can create actions composed of commands, which will be used in the script to execute:

- Basic test functions
- Calculation
- CRM actions (record injection, appointment, reverse annuary, record suppression)
- SQL data access
- Navigation actions (Go to URL, Go to script page)
- Telephony functions (dial a number, change a status, hang up, treat next call, put in queue, play a message, play DTMF, record conversation, call IVR script, show or hide buttons, transfer call, send location to CTI)
- User JavaScript functions
- VMC actions (send e-mail, SMS or fax; perform print and document merge)

« When you decide to load an action template, you can choose one or more commands that make up the action. »

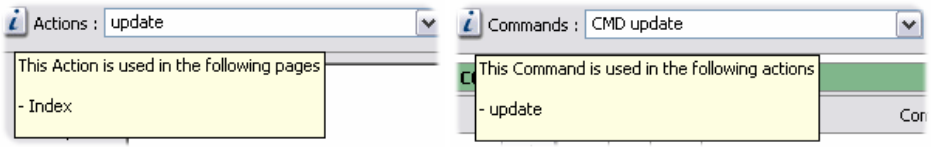


In the **Select or Create an Action** zone, you can manage different actions.



In this box, the user can create a new action, remove an existing one, record modifications and comments, and save an action as template for other users.

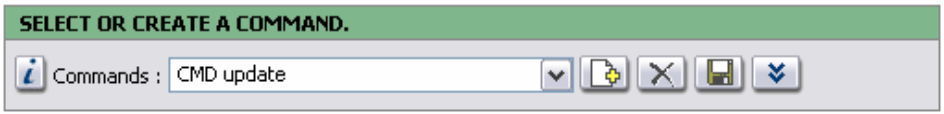
Place the mouse pointer on the symbol to see where the current action or command is called. (Actions tip gives page name, Commands tip give action name).



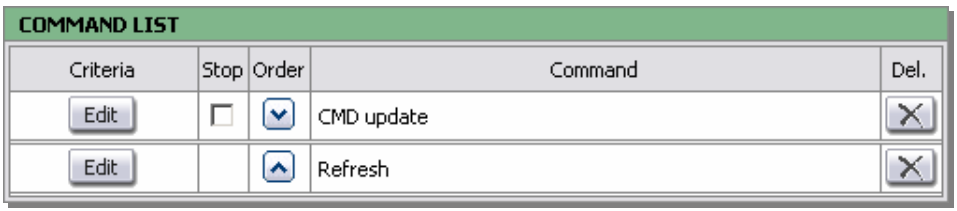
If the symbol is red ( ) , the item is never used. Don't hesitate to delete not used actions, it allows increasing script execution time.

In the next zone, you can view the content of the highlighted action (a list of commands).

In the zone **Select or Create a command**, it is possible to create a new command, remove an existing one, save modifications and add highlighted commands in the action content using the button.



The **Command List** zone allows viewing of the commands the highlighted action consists of.



With the

button related to each command, you can compose criteria that are used to determine whether the command in question must be executed. These criteria are built with the « Criteria Builder » screen. (see chapter « Criteria Builder » on page 140).

If the criteria result is false, then the related command will not be executed.

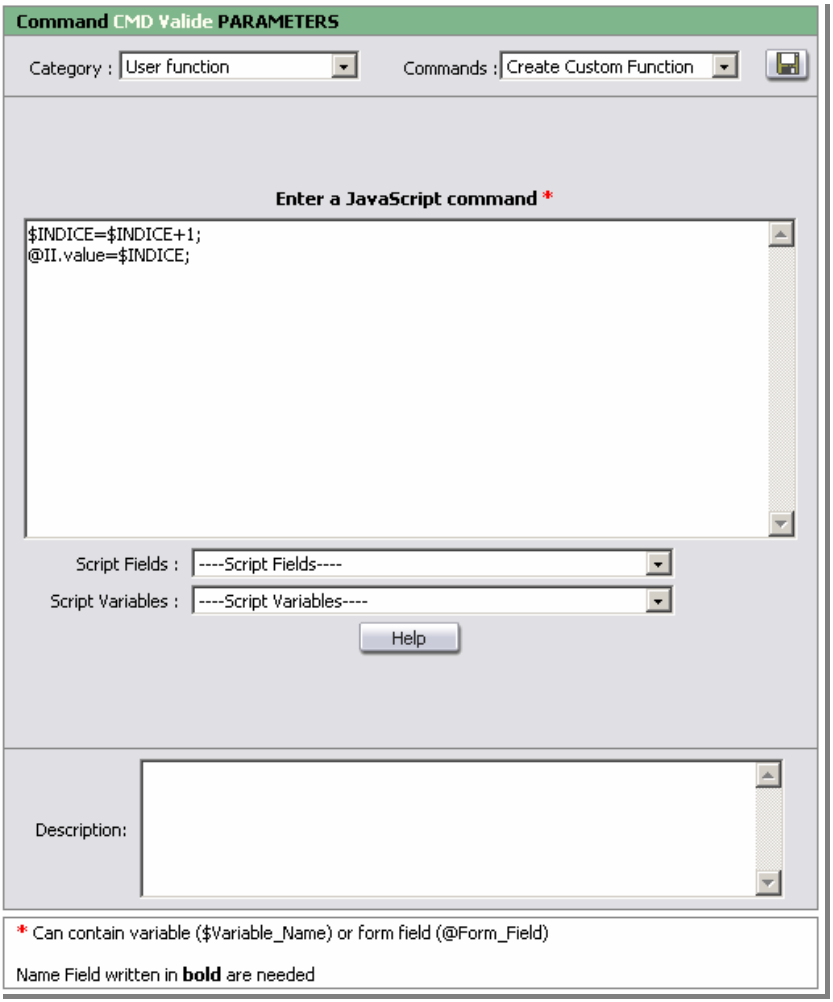
The checkbox in the **Stop** column allows indicating that the commands sequence can be stopped at this location if the criterion result is False.

The and buttons in the **Order** column can be used to change the commands' execution order.

The button can be used to remove a command from the action content.

When a command is highlighted, its content is displayed on the right side of the screen.

In the next example, the command called « CMD Valide » is composed of a **user function** detailed in the zone **Enter Javascript command**. (see section « Description of the possible commands » on page 157)


Command CMD Valide PARAMETERS
Category : User function
Commands : Create Custom Function
Enter a JavaScript command *
$INDICE=$INDICE+1;
@II.value=$INDICE;
Script Fields : ----Script Fields----
Script Variables : ----Script Variables----
Help
Description:
* Can contain variable ($Variable_Name) or form field (@Form_Field)
Name Field written in bold are needed

At the bottom of the screen, a **Description** zone allows typing in a description for the command.

6.8.1 Description of the possible commands

Commands are grouped in several categories.
Each category suggests predefined operations and, for each operation, a specific instruction screen.

Basic Function
This category allows executing a basic test on a specific value according to a format. You can also enter a warning message to inform the user that his data were not formatted correctly.

Calcul (calculate)
In this category you can type in formulas using object values and global variables (Create custom calcul) or you can call a JavaScript function. (Select Template Calcul).

CRM
Perform basic actions on customer records, including:

Record re-injection:
Appointment:
Reverse Annuary:
Clear record: to remove the current customer record.

Data Access

In this category you can define SQL queries on a specific connection. (Select, update, delete, insert, save data form).
You can type in SQL query content in a specific area.

Navigation

This category allows you to perform the following actions:

- Go to page
- Go to URL
- Go to page (in a pop-up with a defined size)
- Go to URL (in a pop-up with a defined size)

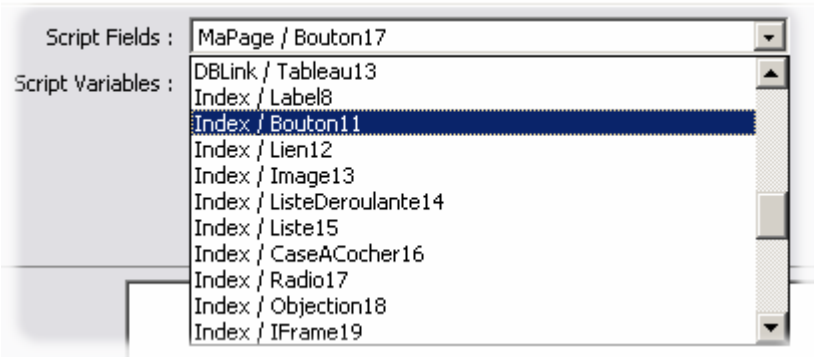
Telephony function

This category allows performing basic telephony tasks:

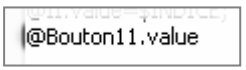
- Dial a number
- Change a status and indicate whether the call was argued positively
- Hang up
- Treat the next call
- Put the current call on hold (waiting queue)
- Play an audio file (see chapter « Upload Center » on page 135)
- Play DTMF keys
- Start recording the current communication
- Call an IVR script
- Show or hide agent toolbar buttons
- Transfer call
- Send location to CTI

User function

This category allows to type in JavaScript code using object values and global variables (Create Custom Function).
The two lists under the input zone let you select an object reference or a variable name to append it to the code.
(see sections « Objects References » and « Definition of global variables » on page 160 and 160)



The first list **Script Fields** gives the object's name and location page. When an object is selected from this list, its reference is attached to the code displayed in the zone above.



The global variable can be added to the code using the second list **Script Variables**.

We can also indicate the name of an existing JavaScript function. This function must first be uploaded using the Upload Center. (see chapter « Upload Center» on page 129) (Select Template Function).

VMC (Vocalcom Media Center)

With this category you can use different communication media such as e-mail, SMS, fax or document print and merge.
According to the selected channel, the screen indicates the fields required to compose the content of the message, and the connection to be used for the document merge.

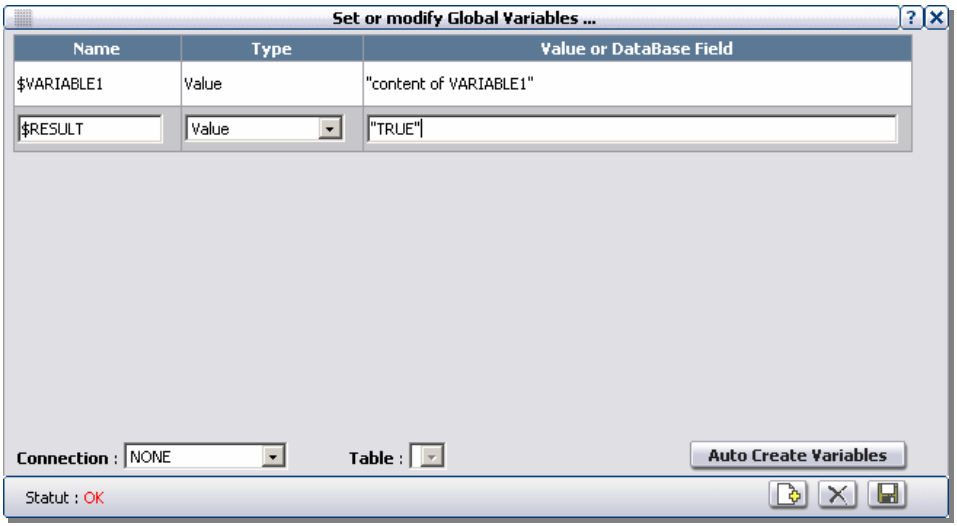
6.8.2 Objects References

All script objects can be added to code or to content of other objects using reference **@< object name>.value**.
For Listbox, DropDownList or Objection objects, this formula gives the content value rather than the displayed text.

The content of a Label object is referred to by the formula **@< label name>.innerHTML**
The background color of a page is referred to by the formula **document.body.bgColor.**
For example: document.body.bgColor = "#FF0000" ;
The page title is referred to by the formula **document.title**.
For example: document.title = "My Test Page" ;

6.8.3 Definition of global variables

In the **Modules** menu, the function **Set Global Variables** can be used to open the following dialog window:



Each variable is displayed on a line by a **Name**, a **Type** and a content **Value or DataBase Field**.

To create a new variable, click on the button to show a new line. Enter a name in the first column.

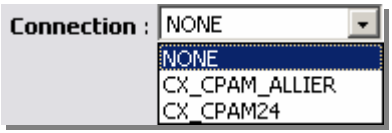
In the **Type** column, select if the variable is a simple value (Value) or if it's linked to a database field. (Bind to database).

If you choose "Value", just enter the content in the 3rd column. Click on the button to validate.

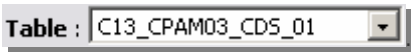
If you choose "Bind to database", the next fields are shown to select a connection, a table and a column.

The button allows creating automatic variables linked to DB fields :

Select a connection in the field **Connection** at the bottom of the dialog window.



Afterwards, select a table within this database via the field **Table**.



Click on the button .

Each table field will be used to create a variable with the same name.

Example :

$INDICE	Attaché à la base	NewComp DB	Constructor	INDICE
$Constructor_Id	Attaché à la base	NewComp DB	Constructor	Constructor_Id
$Constructor_Name	Attaché à la base	NewComp DB	Constructor	Constructor_Nam

The table **Constructor** within the DB **NewComp DB** has 3 fields : INDICE, Constructor_Id and Constructor_Name.

3 variables have been created : $INDICE, $Constructor_Id and $Constructor_Name.

Use the button to delete a variable.

In all the languages, there are reserved words that can not be used as variables. In this case, it's the JavaScript standard reserved words :

boolean	for	return

break	function	short
byte	goto	static
case	if	super
catch	implement	switch
char	import	synchronized
class	in	this
const	instanceof	throw
continue	int	throws
default	interface	transient
do	long	true
double	native	try
else	new	var
extends	null	void
false	package	while
final	private	with
finally	protected	

6.9 Examples of useful scripts

6.9.1 Greeting message according the time

The goal of this example is to display a dynamic message on the first page of the script, according the current time.
The result will be :

"Goodmorning" before 12h00
"Goodafternoon" between 12h00 and 17h00
"Good evening" after 17h00

1. Create a global variable $salutation
2. Type the text on the screen into a textbox, and include $salutation in this text.
3. Create an action **OnLoadPage** composed of a command called **SetSalutation**. Define the content of this command **SetSalutation** with type **User function (Custom function)**. Type the next code:

```
var mytime,hh;
mytime = new Date();
hh = mytime.getHours();
if (hh < 12) { $salutation="Goodmorning";}
if (hh >= 12  && hh < 17) { $salutation="Goodafternoon";}
if (hh >=17) {$salutation="Good evening";}
```

This function extracts the hour from the current date/time value, and according to this hour, set the content of $salutation.

4. Assign the action **OnLoadPage** to the event **OnLoad** of the page.

6.9.2 Use agent identification information

The predefined variable $NAME contains the firstname, the name and the station identification of the agent, with the next format : <Firstname> <Name>, <Station>
Example :

```
Jean-Pierre Petit,Station1
```

On the page, the textbox is defined :

```
$NAME
Only the name of the agent : $AgentName
Only the firstname of the agent : $AgentFirstname
```

The next code can be used to extract the name and firstname .

```
$AgentName = $NAME.substring(0,$NAME.indexOf(','));
$AgentFirstname = $AgentName.substring(0,$AgentName.indexOf(' '));
```

Explanation:
The variable $AgentName contains the part of the $NAME content found before the comma : the firstname and the name.
The variable $AgentFirstname contains the part of the $AgentName content found before the space : the firstname.
The result is :

```
Jean-Pierre Petit,Station1
Only the name of the agent : Jean-Pierre Petit
Only the firstname of the agent : Jean-Pierre
```

Remark 1 : cases of the variables names and functions names (substring, indexOf,...) are important and must be respected.

Reark 2 : previous versions of the scripter don't use all the same predefined variables ($NAME replace previous $TV). Be sure you are using the right documentation and variables version.

6.9.3 Change the mouse pointer

Just like in HTML, it's possible to adapt the mouse pointer when it is over a specific text part.

Into the text, add the next tags :

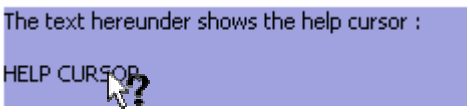
**** and **** arround this specific text.

Example :

```
The text hereunder shows the help cursor :

<font style="cursor:help"> HELP CURSOR </font>
```

Result :



The available cursors are :

cursor:crosshair
cursor:pointer
cursor:move
cursor:text
cursor:wait
cursor:help

cursor:e-resize
cursor:w-resize
cursor:n-resize
cursor:s-resize
cursor:ne-resize
cursor:nw-resize
cursor:se-resize
cursor:sw-resize
cursor:defaut

6.9.4 Set objects invisible or enabled

Using a customized function with the next code, an object can be hidden or shown :

To hide :

@<object name>.style.display='none';

To show :

@<object name>.style.display='inline';

A button can be set disabled using :

@<button name>.disabled = true ;

Or enabled :

@<button name>.disabled = false ;